BRUNO ZANETTI MELOTTI

EFEITO DO RANKING SOBRE MÉTRICAS DE CATEGORIZAÇÃO MULTI-RÓTULO DE TEXTO

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do Grau de Mestre em Engenharia Elétrica.

Orientador: Prof. Dr. Alberto Ferreira De Souza

VITÓRIA
2009

BIBLIOTECA

BRUNO ZANETTI MELOTTI

EFEITO DO RANKING SOBRE MÉTRICAS DE CATEGORIZAÇÃO MULTI-RÓTULO DE TEXTO

Dissertação submetida ao programa de Pós-Graduação em Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para a obtenção do Grau de Mestre em Engenharia Elétrica.

Aprovada em 27 de novembro de 2009.

COMISSÃO EXAMINADORA

Prof. Dr. Alberto Ferreira De Souza - Orientador
Universidade Federal do Espírito Santo

Prof. Dr. Claudine Badue Gonçalves
Universidade Federal do Espírito Santo

Prof. Dr. Felipe Maia Galvão França
Universidade Federal do Rio de Janeiro

EPÍGRAFE

"Uma jornada de duzentos quilômetros começa com um simples passo" (Provérbio Chinês)

DEDICATÓRIA

Dedico este trabalho aos meus pais Silvério e Zenir e aos meus irmãos Breno e Lorena pelo amor, pela compreensão, pelas lições de vidas e pelo apoio dado ao longo desses anos de vida.

A minha noiva Viviani pelo companheirismo, pelo amor e pelo apoio para a realização dos nossos objetivos, em especial este trabalho.

AGRADECIMENTOS

Agradeço, primeiramente, a Deus pela oportunidade de realização deste trabalho, e depois a minha família e a minha noiva pelo apoio e incentivo.

À equipe do SCAE de pesquisadores pelo apoio nos estudos e aos programadores, em especial ao meu amigo Felipe Thomaz Pedroni.

À Receita Federal do Brasil pelo apoio financeiro para a realização do projeto SCAE, e conseqüentemente deste trabalho.

Ao meu orientador Prof. Dr. Alberto de Souza, por qual tenho muito apreço, pela orientação, pelo acompanhamento, pela atenção e pela paciência ao longo do desenvolvimento deste trabalho.

Ao Laboratório de Computação de Alto Desempenho (LCAD) por disponibilizar um *cluster* para a realização dos experimentos deste trabalho e a Renderson Loriato da Silva pela manutenção do *cluster*.

Agradeço à Prof. Dr. Claudine Badue pelo apoio durante a realização deste trabalho.

E a todos os meus amigos e pessoas que contribuíram de forma direta ou indireta para realização deste projeto.

RESUMO

Dado um documento para categorização, um sistema de categorização multi-rótulo de texto tipicamente ordena um conjunto de categorias pré-definido, de acordo com a adequação delas ao documento, e seleciona as categorias do topo do *ranking* como o conjunto de categorias do documento. Empates no *ranking* eventualmente existentes podem ser tratados de diferentes maneiras, mas, muito embora isso possa afetar as métricas utilizadas para avaliar o desempenho dos categorizadores multi-rótulo de texto, este problema parece ter sido pouco estudado na literatura. Neste trabalho, analisamos o impacto de diferentes tipos de *ranking* sobre diversas métricas de avaliação de desempenho de categorizadores multi-rótulo de texto, a saber: *one-error, coverage, ranking loss, average precision, R-precision, Hamming loss, exact match, precision, recall, e* F_1. Para isso, reformulamos sua definição de modo a considerar empates de acordo com o tipo de *ranking* empregado. Utilizamo-las então para avaliar o desempenho das técnicas de categorização multi-rótulo de texto k-vizinhos mais próximos (k*NN*), k-vizinhos mais próximos multi-rótulo (*ML-*k*NN*), rede neural sem peso do tipo *VG-RAM* (*VG-RAM WNN*) e VG-RAM com correlação de dados (*VG-RAM WNN-COR*) na categorização de duas bases multi-rótulo de texto com grande número de categorias (105 e 692 categorias). Descobrimos que, dependendo do tipo de *ranking* empregado, os resultados de desempenho são significativamente diferentes para muitas das métricas analisadas, o que sugere que o tipo de *ranking* deve ser claramente indicado na avaliação de técnicas de categorização multi-rótulo de texto.

ABSTRACT

A multi-label text categorization system typically ranks a set of predefined labels according to their appropriateness to a given document and then selects the top ranking labels as the document's label set. Ties occurring in the ranking can be broken in many different ways but, although this may affect the metrics used to evaluate the multi-label text categorizer, the issue seems to have been little addressed in the literature. In this paper, we analyze the impact of different ranking methods on ten multi-label text categorization performance metrics: one-error, coverage, ranking loss, average precision, R-precision, Hamming loss, exact match, precision, recall, and F_1. To this end, we first reformulate some of the metrics in order for ties to be taken into account. We then use them to evaluate the performance of three multi-label text categorization techniques, k-nearest neighbors (kNN), multi label k-nearest neighbors (ML-kNN), virtual generalizing random access memory weightless neural networks (VG-RAM WNN) and VG-RAM Data Correlation (VG-RAM WNN-COR), on the categorization of two multi-label text databases with large numbers of labels (105 and 692 categories). We have found that, depending on the method adopted for ranking, the performance results are significantly different for many of the metrics in question, which suggests that the particular ranking method one uses should always be indicated clearly whenever evaluating multi-label text categorization techniques.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

1 INTRODUÇÃO

A convergência da computação e da comunicação, principalmente a disseminação da Internet, tem produzido uma sociedade que se alimenta de informação, cuja maior parte está na forma bruta: dados. A quantidade de dados no mundo, em nossas vidas, no meio corporativo (comércio e indústria) e nas instituições governamentais cresce exponencialmente e sem fim à vista. Com isso, hoje há uma grande lacuna entre a geração de dados e a compreensão deles. Transformar dados em informação e informação em conhecimento é um dos grandes desafios atuais [Sbc09].

O aumento do volume de dados sob a forma de texto tem provocado um grande interesse no desenvolvimento de técnicas capazes de tornar a organização e a gestão da informação tarefas eficientes e rápidas. Separar a informação em um conjunto pré-definido de categorias facilita a organização, gestão, recuperação e categorização de informação relevante, e em muitas empresas e instituições governamentais, profissionais são treinados para tal tarefa. Categorizar texto segundo um conjunto pré-definido de categorias, como é feito, por exemplo, em jornais e revistas, que agrupam conteúdo em seções (categorias) pré-definidas, é hoje um importante mecanismo de organização da informação para o consumo humano. Contudo, a categorização manual é um processo demorado e custoso, o que limita sua aplicabilidade, pois, alterando o contexto do problema, novas regras de categorização precisam ser definidas manualmente. Conseqüentemente, existe um grande interesse no meio acadêmico e corporativo em desenvolver técnicas para categorização automática de texto [Sebastiani02]. No entanto, a categorização automática de texto ainda é um problema computacionalmente desafiador para a comunidade de Recuperação de Informação (RI), tanto no contexto acadêmico quanto no industrial.

A maior parte dos trabalhos sobre categorização de texto na literatura está focada na categorização de texto com um úni-rótulo (*single-label*), onde exatamente uma categoria deve ser atribuída a cada documento [Sebastiani02]; por exemplo, na filtragem de *spam*, uma mensagem de e-mail deve ser categorizada nas categorias *spam* ou não *spam*. No entanto, em problemas do mundo real, a categorização multi-rótulo (*multi-label*) é freqüentemente necessária [McCallum99, Schapire00, Clare01b, Elisseeff02, Comité03, Ueda03, Boutell04, Kazawa05, Zhang06, Zhang07]. Diversas técnicas têm sido propostas para tratar o problema de categorização multi-rótulo, tais como: árvores de decisões multi-rótulo (*multi-label*

decision trees) [Clare01a, Comité03], métodos *kernel* (*kernel methods*) [Elisseeff02, Boutell04, Kazawa05], ou redes neurais (*neural networks*), e muitas delas para categorização multi-rótulo de texto (*multi-label text categorization*) [Gao04, McCallum99, Romero04, Schapire99, Badue08, Ciarelli09, DeSouza08, DeSouza09a, DeSouza09b, Oliveira08a, Oliveira08b, Ueda03, Yang01].

Tipicamente, para realizar a tarefa de categorização, um sistema de categorização multi-rótulo de texto inicialmente precisa ser treinado a partir de documentos previamente categorizados em uma ou mais categorias de um conjunto pré-definido de categorias. Cada sistema de categorização, de acordo com a técnica empregada, explora de uma forma distinta as características dos documentos e categorias associadas. Baseado nessas características, dado um novo documento, o sistema de categorização multi-rótulo atribui um valor real entre zero e um a cada categoria – este valor indica o grau de crença com que o sistema atribui cada categoria (rótulo) ao documento. Com estes valores pode ser construída uma lista das categorias que o sistema crê que devem ser atribuídas ao novo documento. Normalmente, essa lista é ordenada em ordem decrescente do grau de crença atribuído pelo categorizador a cada uma das categorias, formando um *ranking* de categorias. Esse *ranking* é semelhante ao *ranking* de uma competição, onde o primeiro colocado está na primeira posição do *ranking*, o segundo colocado na segunda posição, e assim por diante.

Várias métricas de avaliação de desempenho de categorizadores multi-rótulo propostas na literatura de RI são baseadas no *ranking* de categorias produzido pelos categorizadores para cada documento. Contudo, as definições dessas métricas, presentes na literatura, não consideram os casos em que o sistema de categorização é incapaz de evitar empates entre categorias, isto é, atribuir o mesmo grau de crença a diferentes categorias no *ranking* [Schapire99, Schapire00, Sebastiani02, Kazawa05, Manning08].

A Tabela 1.1 mostra um *ranking* com empates produzido por um categorizador hipotético para um documento que pode ser categorizado em 100 diferentes categorias $C = \{c_1, c_2 \ldots, c_{100}\}$. Na tabela, a coluna *Categoria* mostra as categorias atribuídas ao documento e a coluna *Crença* mostra o grau de crença do categorizador de que a categoria deve ser atribuída ao documento. Note, na tabela, que a ordenação entre as categorias c_2 e c_3 é arbitrária, pois elas estão empatadas. Suponha que o conjunto de categorias corretas para este documento seja $\{c_1, c_2\}$. Dependendo de como a ordenação de c_2 e c_3 é tratada no *ranking*, a saída do categorizador pode ser $\{c_1\}$, $\{c_1, c_2\}$, $\{c_1, c_3\}$ ou $\{c_1, c_2, c_3\}$, e, de acordo com muitas métricas de avaliação multi-rótulo, o desempenho do categorizador será diferente

para cada um desses conjuntos de saída. Na verdade, se o conjunto de categorias correto for $\{c_1, c_2, c_4\}$, o valor de algumas métricas pode variar consideravelmente devido ao grande número de categorias empatadas – na Tabela 1.1, 95 rótulos possuem a mesma crença de c_4 (zero).

Tabela 1.1 - Ordenação com empates na saída do categorizador hipotético.

Categoria	Crença
c_1	*0,5*
c_2	*0,2*
c_3	*0,2*
c_8	*0,1*
$\{c_4, \ldots, c_{100}\}$	*0,0*

O tratamento de empates na avaliação de desempenho de categorizadores ainda é um tópico pouco tratado pela literatura de RI. Dentre os poucos pesquisadores que investigaram o assunto ou temas correlatos, Cooper, em 1968 [Cooper68], propôs uma métrica chamada Tamanho da Procura Esperada (*Expected Search Length* - *ESL*) para avaliar os *rankings* produzidos pelas ferramentas de buscas de documentos. A métrica *ESL* considera que empates no *ranking* podem existir. Contudo, até onde pudemos examinar, a métrica *ESL* é equivalente à métrica *coverage* (ver Seção 4.1.2, pág. 69). Fagin, em 2004 [Fagin04], propôs a técnica de agregação de *rankings* (*ranking agregation*) para tratamento de empates no contexto de ferramentas de busca. Dados múltiplos *rankings*, a técnica proposta cria um único *ranking* que minimiza a distância Kendall-tau [Fagin03]. Entretanto, essa técnica não pode ser apropriadamente adaptada para avaliar o desempenho de categorizadores segundo as métricas correntemente em uso pela comunidade de RI, uma vez que estas métricas não são baseadas na comparação de *rankings*. Em 2006 [Fagin06], Fagin definiu quatro tipos de métricas para comparar *rankings* com empates baseadas na generalização da distância Kendall-tau e a distância Spearman *footrule* [Fagin03]. Novamente, essas métricas não são apropriadas para avaliar o desempenho de categorizadores, mas sim comparar o desempenho de ferramentas de busca (são baseadas na comparação de *rankings*).

Neste trabalho examinamos o efeito de diferentes tipos de *ranking* sobre as métricas mais populares de categorização multi-rótulo de texto empregadas pela comunidade de RI. Avaliamos experimentalmente os efeitos dos tipos de *ranking* Ordinal Aleatório, Denso, Padrão e Modificado [Wikipedia09] nas seguintes métricas de avaliação de desempenho de categorizadores multi-rótulo: *one-error* [Schapire99], *coverage* [Schapire00], *ranking loss* [Schapire99], *average precision* [Schapire00, Manning08], *R-precision* [Manning08], *Hamming loss* [Schapire99], *exact match* [Kazawa05], *precision* [Sebastiani02], [Manning08], *recall* [Sebastiani02, Manning08], e F_1 [Sebastiani02, Manning08]. Para isso, avaliamos, usando as métricas mencionadas, o desempenho dos categorizadores k-vizinhos mais próximos (*k-nearest neighbors*) (*kNN*) [Mitchell97, Baoli03, Hao07], k-vizinhos mais próximos multi-rótulo (*multi-label k-nearest neighbors*) (*ML-kNN*) [Zhang07], e rede neural sem peso do tipo VG-RAM (*virtual generalizing random access memory weightless neural networks* – VG-RAM WNN) [Aleksander98, Badue08, DeSouza07, DeSouza08, DeSouza09a, DeSouza09b] no contexto da categorização de descrições de atividades econômicas de empresas brasileiras segundo a Classificação Nacional de Atividades Econômicas (CNAE) [CNAE03]. Nossos resultados mostraram que, dependendo do tipo de *ranking* utilizado, o desempenho dos categorizadores pode ser significativamente diferente para muitas das métricas examinadas.

1.1 Motivações

Quando existem empates no *ranking*, as categorias podem ser ordenadas de várias maneiras. Entretanto, as métricas de avaliação de desempenho de categorizadores multi-rótulo de texto propostas na literatura desconsideram a existência de empates no *ranking*. Assim, a principal motivação para o desenvolvimento deste trabalho foi avaliar o impacto dos empates no desempenho dos categorizadores multi-rótulo medido segundo as métricas mais populares da literatura.

A motivação para este trabalho surgiu durante o desenvolvimento do Sistema Computacional de Codificação Automática de Atividades Econômicas (SCAE). Tal sistema se propõe a categorizar automaticamente, segundo a CNAE, descrições, na forma de texto livre, de atividades econômicas de empresas brasileiras.

A CNAE lista todas as atividades econômicas legalmente reconhecidas no Brasil. Correntemente, a CNAE contempla 1.301 atividades econômicas, cada uma possuindo um código específico. Empresas podem ser categorizadas dentro de um ou mais códigos; ou seja, categorizar empresas segundo a CNAE é um problema de categorização multi-rótulo. Devido à grande quantidade de categorias, este é um problema complexo e incomum na literatura.

Nossos estudos com o SCAE mostraram que, especialmente com um número grande de categorias, empates no *ranking* produzido pelos mais diversos categorizadores são freqüentes, o que torna relevante o tratamento destes empates no contexto da avaliação do desempenho destes categorizadores.

1.2 Objetivos

Diferentes tipos de *ranking* tratam empates de diferentes formas. Por essa razão, o objetivo principal deste trabalho foi (i) examinar experimentalmente o impacto do emprego dos *rankings* Ordinal Aleatório, Denso, Padrão e do Modificado no desempenho de categorizadores multi-rótulo de texto medido com as métricas mais populares de avaliação de desempenho de categorizadores multi-rótulo: *one-error*, *coverage*, *ranking loss*, *average precision*, *R-precision*, *Hamming loss*, *exact match*, *precision*, *recall* e F_1. A formulação original de algumas destas métricas não comporta os tipos de *ranking* Denso, Padrão ou Modificado. Assim, também foi objetivo deste trabalho (ii) reformular a definição das métricas que não comportam esses tipos de *ranking*. Além desses objetivos, neste trabalho buscamos (iii) identificar a forma de raqueamento mais apropriada para o tratamento de empates no contexto de cada métrica.

1.3 Contribuições

As principais contribuições deste trabalho foram:

- A demonstração experimental de que, dependendo do tipo de *ranking* utilizado, o desempenho dos categorizadores multi-rótulo são significativamente diferentes para muitas das métricas examinadas, o que mostra que, quando técnicas de

categorização multi-rótulo são avaliadas, o método de *ranking* empregado deve ser indicado.

- A reformulação (generalização) de várias métricas de avaliação de desempenho de categorizadores multi-rótulo para comportar o tratamento de empates observados nos *rankings* Denso, Padrão e Modificado.
- A identificação do tipo de *ranking* apropriado para tratamento de empates para cada métrica de avaliação de categorizadores multi-rótulo examinada.

1.4 Publicações

1.4.1 Revistas

- Alberto F. De Souza, Bruno Zanetti Melotti, Claudine Badue. *Multi-Label Text Categorization with a Data Correlated VG-RAM Weightless Neural Network*. In International Journal of Computational Intelligence Research (IJCIR), 2009 (*accepted for publication*).

1.4.2 Conferências

- Alberto F. De Souza, Claudine Badue, Bruno Zanetti Melotti, Felipe T. Pedroni, Fernando Líbio L. Almeida. *Improving VG-RAM WNN Multi-label Text Categorization via Label Correlation*. In: 2nd Workshop on Intelligent Text Categorization and Clustering (WITCC'08), 8th International Conference on Inteligent System Design and Applications (ISDA'08), Kaohsiung City, Taiwan, 2008.

1.5 Organização deste trabalho

Após esta introdução, esta dissertação está organizada da seguinte forma:

- O Capítulo 2 apresenta uma definição formal de categorização multi-rótulo de texto, os tipos de *rankings* examinados, as técnicas de categorização multi-rótulo utilizadas neste trabalho e, por fim, o problema de categorização de atividades econômicas;
- O Capítulo 3 descreve a metodologia empregada nos experimentos para a avaliação do impacto dos tipos de *ranking* no desempenho dos categorizadores multi-rótulo examinados;
- O Capítulo 4 demonstra experimentalmente o efeito dos tipos de *ranking* no desempenho dos categorizadores;
- O Capítulo 5 apresenta uma discussão dos resultados obtidos, trabalhos correlatos e uma análise crítica deste trabalho;
- Por fim, o Capítulo 6 sumariza os resultados obtidos e apresenta as nossas conclusões e propostas de trabalhos futuros.

2 CATEGORIZAÇÃO MULTI-RÓTULO DE TEXTO

Neste Capítulo, formalizamos o conceito de categorização multi-rótulo de texto e definimos os tipos de *ranking* Ordinal Aleatório, Denso, Padrão e Modificado empregados para avaliar o impacto dos mesmos no desempenho dos categorizadores multi-rótulo de texto k-vizinhos mais próximos (k-*nearest neighbors*) (k*NN*) [Mitchell97, Baoli03, Hao07], k-vizinhos mais próximos multi-rótulo (*multi-label* k-*nearest neighbors*) (*ML-*k*NN*) [Zhang07] e rede neural sem peso do tipo *VG-RAM* (*virtual generalizing random access memory weightless neural networks – VG-RAM WNN*) [Aleksander98, Badue08, DeSouza07, DeSouza08, DeSouza09a, DeSouza09b]. Apresentamos, também, o domínio do problema de descrições de atividades econômicas de empresas brasileiras segundo a Classificação Nacional de Atividades Econômicas (CNAE) [CNAE03], e como essas descrições de atividades são representadas internamente, segundo o modelo vetorial [Salton75, Baeza99], nas técnicas de categorização multi-rótulo de texto.

2.1 Categorização multi-rótulo de texto

Sejam D um domínio de documentos e $C = \{c_1, \ldots, c_{|C|}\}$ um conjunto de categorias pré-definido. Na categorização multi-rótulo de texto, os documentos de D podem ser categorizados dentro de uma ou mais categorias de C.

Seja $\Omega = \{d_1, \ldots, d_{|\Omega|}\} \subset D$ um *corpus* inicial de documentos previamente categorizados manualmente por especialistas no domínio dentro de subconjuntos de C. Em sistemas automáticos para categorização multi-rótulo, um subconjunto de Ω, denominado conjunto de treinamento (e validação), $TV = \{d_1, \ldots, d_{|TV|}\}$, pode ser utilizado para treinar (e validar) categorizadores implementados segundo técnicas de aprendizado de máquina [Sebastiani02] (neste trabalho empregamos somente categorizadores automáticos baseados em técnicas de aprendizado de máquina). O conjunto de teste, $Te = \{d_{|TV|+1}, \ldots, d_{|\Omega|}\} = \Omega - TV$, por outro lado, consiste dos documentos não empregados no treinamento dos sistemas de categorização e somente submetidos a estes em fase de testes. Depois de ser treinado (e

calibrado) com TV, um sistema de categorização pode ser utilizado para predizer o conjunto de categorias de cada documento em Te.

Sistemas automáticos para categorização multi-rótulo tipicamente implementam uma função $f : D \times C \rightarrow \Re$ que retorna o grau de crença para cada par $\langle d_j, c_i \rangle \in D \times C$, ou seja, um número entre 0 e 1 que, a grosso modo, representa evidência de que o documento de teste d_j deve ser categorizado dentro da categoria c_i. A função $f(.,.)$ pode ser transformada em uma função ranqueadora $r(.,.)$, tal que, se $f(d_j, c_i) > f(d_j, c_k)$, então $r(d_j, c_i) < r(d_j, c_k)$, e se $f(d_j, c_i) < f(d_j, c_k)$, então $r(d_j, c_i) > r(d_j, c_k)$; mas e se $f(d_j, c_i) = f(d_j, c_k)$?

Seja C_j o conjunto de categorias pertinentes (categorias especificadas pelos especialistas no domínio) ao documento de teste d_j e $\hat{C}_j$ o conjunto de categorias preditas para d_j por um categorizador automático. Um bom categorizador automático tenderá a posicionar as categorias de C_j em posições mais elevadas no *ranking* do que aquelas que não pertencem a C_j. As categorias c_i cujo grau de crença é superior ao limiar de corte τ_i são preditas para o documento de teste d_j, isto é, $\hat{C}_j = \{c_i \mid f(d_j, c_i) \geq \tau_i\}$.

Quando não existem empates no *ranking*, isto é, $f(d_j, c_i) \neq f(d_j, c_k)$ para qualquer $i \neq k$, a função $f(.,.)$ pode ser transformada em uma função ranqueadora $r(.,.)$, que produz um *ranking* que é um mapeamento um-para-um sobre $\{1,2,\ldots,|C|\}$. Entretanto, se existem empates ($f(d_j, c_i) = f(d_j, c_k)$ para $i \neq k$), as categorias podem ser ranqueadas de várias formas diferentes.

2.2 Tipos de *ranking*

Um *ranking* é uma relação entre um conjunto de elementos tal que, para quaisquer dois elementos, o primeiro é colocado numa posição superior, inferior ou igual em relação ao segundo de acordo com um determinado critério [Wikipedia09]. Nem sempre é possível atribuir um único elemento para cada posição do *ranking*. Por exemplo, em uma competição, dois (ou mais) participantes podem empatar em alguma posição do *ranking*. Do mesmo modo, em uma tarefa de categorização multi-rótulo de texto, um categorizador pode atribuir o mesmo grau de crença a mais de uma categoria. Neste caso, vários tipos de *ranking* podem ser

adotados e o tipo de *ranking* pode afetar o resultado final de uma avaliação de desempenho do categorizador.

Um possível tipo de *ranking*, que chamamos de *ranking* Ordinal Aleatório (*Ordinal ranking*) (Figura 2.1(a)) [Wikipedia09], atribui posições distintas no *ranking* para todas as categorias, incluindo aquelas empatadas. Neste tipo de *ranking*, a atribuição de posições distintas para as categorias empatadas é feita aleatoriamente. Acreditamos que este tipo de *ranking* é o empregado, senão em todos, na grande maioria dos trabalhos sobre categorização multi-rótulo de texto da literatura. Um segundo tipo de *ranking*, chamado *ranking* Denso (*Dense ranking*) (Figura 2.1(b)) [Wikipedia09], atribui a mesma posição no *ranking* para as categorias com mesmos valores de $f(d_j,.)$. Um terceiro tipo de *ranking*, chamado *ranking* Padrão (*Standard competition ranking*) (Figura 2.1(c)) [Wikipedia09], atribui também a mesma posição do *ranking* para categorias empatadas. Entretanto, neste *ranking*, lacunas são deixadas nas posições do *ranking* após cada conjunto de categorias com valores iguais de $f(d_j,.)$. O número de posições no *ranking* que são deixadas vazias é o número de categorias com mesmo grau de crença menos um. Outro tipo de *ranking*, o *ranking* Modificado (*Modified competition ranking*) (Figura 2.1(d)) [Wikipedia09], é muito similar ao *ranking* Padrão. A única diferença é que, no *ranking* Modificado, as lacunas nas posições do *ranking* são deixadas antes de cada conjunto de categorias com o mesmo valor de $f(d_j,.)$.

ORDINAL		
Posição	**Categoria**	**Crença**
1	{c1}	*0,5*
2	{c2}	*0,2*
3	{c3}	*0,2*
4	{c8}	*0,1*

(a) *Ranking* Ordinal

DENSO		
Posição	**Categoria**	**Crença**
1	$\{c_1\}$	*0,5*
2	$\{c_2\}$	*0,2*
2	$\{c_3\}$	*0,2*
3	$\{c_8\}$	*0,1*

(b) *Ranking* Denso

PADRÃO		
Posição	**Categoria**	**Crença**
1	$\{c_1\}$	*0,5*
2	$\{c_2\}$	*0,2*
2	$\{c_3\}$	*0,2*
4	$\{c_8\}$	*0,1*

(c) *Ranking* Padrão

MODIFICADO		
Posição	**Categoria**	**Crença**
1	$\{c_1\}$	*0,5*
3	$\{c_2\}$	*0,2*
3	$\{c_3\}$	*0,2*
4	$\{c_8\}$	*0,1*

(d) *Ranking* Modificado

Figura 2.1 - Exemplos de tipos de *ranking*.

Note que os tipos de *ranking* Ordinal Aleatório, Denso, Padrão e Modificado diferem entre si na forma como as categorias empatadas no *ranking* são ordenadas. Caso não existam empates, os quatro tipos de *ranking* são equivalentes.

2.3 Representação vetorial de documentos

Os documentos, em seu formato original (texto livre), usualmente não podem ser tratados diretamente por técnicas de aprendizado de máquina empregadas na construção de categorizadores automáticos de texto. Na maioria das técnicas de aprendizado de máquina, cada documento do conjunto Ω é representado por um vetor de números na representação ponto-flutuante; esta forma de representação de documentos é conhecida na literatura como representação vetorial de documentos [Baeza99]. Cada elemento deste vetor quantifica a freqüência com que um termo, pertencente a um vocabulário de termos conhecidos pelo categorizador, aparece em *TV* (*bag-of-words representation* – [Baeza99, Sebastiani02]). Um termo é simplesmente uma ou mais palavras cujo significado, ou semântica, é representativo para o documento [Baeza99, Sebastiani02].

Formalmente, no modelo vetorial de representação de documentos [Salton75, Baeza99], os documentos são representados por vetores no espaço $\Re^n$, onde n representa o número de termos do vocabulário de termos conhecidos pelo categorizador. Cada documento d_j do conjunto Ω é representado por um vetor de pesos $\vec{d}_j = \langle w_{1j}, w_{2j}, \ldots, w_{|T|j} \rangle$, onde T é o conjunto dos termos que ocorrem pelo menos uma vez nos documentos de *TV* e w_{kj} representa o peso do termo t_k do documento d_j; a ordem dos termos em $\vec{d}_j$ é a mesma para qualquer *j* [Sebastiani02].

A Figura 2.2 mostra um exemplo de um *corpus* formado pelo conjunto de documentos $\Omega = \{d_1, d_2, d_3\}$, representados vetorialmente por meio de vetores tridimensionais, onde cada dimensão está associada aos pesos dos termos do conjunto $T = \{t_1, t_2, t_3\}$ nos documentos. O documento d_1 é representado pelo vetor $\vec{d}_1 = \langle w_{11}, w_{21}, w_{31} \rangle$, d_2 por $\vec{d}_2 = \langle w_{12}, w_{22}, w_{32} \rangle$ e d_3 por $\vec{d}_3 = \langle w_{13}, w_{23}, w_{33} \rangle$.

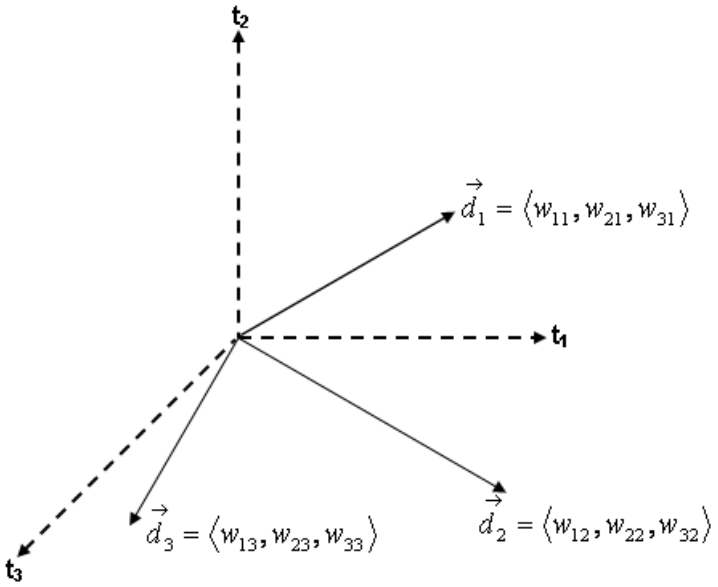


Figura 2.2 - Representação gráfica de três documentos de acordo com o modelo vetorial.

Para determinar o peso w_{kj} do termo t_k no documento d_j, diversas formulações podem ser utilizadas. Empregamos a função de ponderação conhecida como tfidf (*term frequency, inverse document frequency*) [Sebastiani02], definida na Equação (2.1), abaixo.

$$tfidf(t_k, d_j) = \#(t_k, d_j) * \log\left(\frac{|TV|}{\#TV(t_k)}\right) \tag{2.1}$$

onde $\#(t_k, d_j)$ representa o número de vezes que o termo t_k ocorre no documento d_j, chamada de freqüência do termo (*term frequency – tf*); $\#TV(t_k)$ denota o número de documentos do conjunto TV em que o termo t_k ocorre; e o termo $\log\left(\frac{|TV|}{\#TV(t_k)}\right)$ é chamado de freqüência inversa do documento (*inverse document frequency– idf*).

A função *tfidf* codifica a intuição de que (i) quanto mais freqüente um termo em um documento, maior é a importância semântica dele para o documento, e (ii) quanto mais freqüente um termo no conjunto de documentos TV, menor é o poder de discriminação dele. Esta formulação leva em consideração apenas a ocorrência dos termos, não considerando a ordem na qual eles aparecem nos documentos e o papel sintático que eles possuem. É importante observar que os pesos dos termos são mutuamente independentes, isto é, o peso w_{kj} calculado para o par (t_k, d_j) não diz nada a respeito do peso w_{k+1j} calculado para o par (t_{k+1}, d_j) [Baeza99].

Para que os pesos estejam no intervalo [0, 1] e para que os documentos sejam representados por vetores de mesma magnitude, os pesos calculados por *tfidf* são freqüentemente normalizados pela função de normalização de co-seno, definida na Equação (2.2) [Sebastiani02].

$$w_{kj} = \frac{tfidf(t_k, d_j)}{\sqrt{\sum_{s=1}^{|T|} (tfidf(t_s, d_j))^2}} \tag{2.2}$$

O procedimento de transformar os textos dos documentos em uma forma que possa ser interpretada pelas técnicas de categorização de texto é chamado de indexação (*indexing*). A função de indexação *tfidf* foi escolhida para ser utilizada neste trabalho por ser a mais empregada na literatura [Sebastiani02], ou seja, as técnicas de categorização multi-rótulo examinadas neste trabalho têm como entrada documentos (ou descrições de atividades econômicas) representados por vetores cujos pesos dos termos são calculados pela função *tfidf*. Estas técnicas são apresentadas nas seções a seguir.

2.4 Categorizador *kNN*

A técnica *k*-vizinhos mais próximos (*k-nearest neighbor – kNN*) é *instance-based* [Mitchell97], isto é, nenhum modelo é criado para extrair as características de um documento e associá-las a um conjunto de categorias na base de treinamento. Métodos *instance-based* são, algumas vezes, referenciados como métodos de aprendizado “preguiçosos” (*lazy learning methods*) porque eles somente processam a base de treinamento ao receber uma nova requisição de categorização para um novo documento [Mitchell97].

O *kNN* tradicional é utilizado na literatura em contextos úni-rótulo, mas o problema em que estamos interessados é multi-rótulo. Para empregar o categorizador *kNN* em problemas multi-rótulo, ele precisa ser alterado. A Subseção 2.4.1 apresenta o *kNN* úni-rótulo e a Subseção 2.4.2 um *kNN* multi-rótulo desenvolvido no Laboratório de Computação de Alto Desempenho (LCAD) da UFES.

2.4.1 Categorizador *kNN* Úni-rótulo

O categorizador $k\,NN$ é uma abordagem eficiente e amplamente utilizada em problemas de categorização de texto [Baoli03]. A idéia por trás do algoritmo $k\,NN$ é bastante simples: para categorizar o documento de teste d_j, o sistema busca, empregando uma métrica de distância, os k vizinhos (documentos) mais próximos a d_j no conjunto de treinamento e utiliza as categorias desses k vizinhos para predizer a categoria de d_j [Hao07]. Várias métricas de distância podem ser utilizadas [Ciarelli08, Oliveira08b], mas a mais freqüente é o co-seno do ângulo entre o vetor que representa d_j, $\vec{d}_j$, e cada documento *m* de *TV*, $\vec{d}_m$ [Sebastiani02]:

$$\cos(\vec{d}_j, \vec{d}_m) = \frac{\vec{d}_j \bullet \vec{d}_m}{|\vec{d}_j| \times |\vec{d}_m|} \tag{2.3}$$

A saída do categorizador $k\,NN$ são os pares $\langle d_j, c_i \rangle \in D \times C$, e o valores dos co-senos dos ângulos representam o grau de crença do categorizador de que a categoria c_i deve ser atribuída ao documento d_j. A partir deste grau de crença, duas estratégias podem ser utilizadas para predizer o conjunto de categorias: maioria dos votos (*majority votes*) e a soma dos co-senos (similaridades) (*similarity score summing*) [Baoli03, Yavuz98, Yang99]. Na primeira, a categoria mais freqüente entre as dos k vizinhos mais próximos é a escolhida. Na segunda, a categoria com a maior soma dos co-senos é a escolhida.

2.4.2 Categorizador *kNN* Multi-rótulo

Na categorização multi-rótulo, o $k\,NN$, ao invés de predizer um único par $\langle d_j, c_i \rangle$, tem que predizer um conjunto de pares $\langle d_j, c_i \rangle$ (com um ou mais elementos) para o documento de teste d_j. Para isso, o $k\,NN$ multi-rótulo busca, empregando uma métrica de distância, os k documentos mais próximos de d_j no conjunto de treinamento. A medida de distância entre d_j e cada um dos k documentos é usada como medida de crença do

categorizador de que as categorias associadas a cada documento devam ser preditas para d_j. Isto é, a crença de que uma categoria c_i associada a d_m deva ser atribuída a d_j, $f(d_j, c_i)$, é igual a $\cos(\vec{d}_j, \vec{d}_m)$, para c_i pertencente a C_m e 1 <= m <= k. Pode haver mais de uma ocorrência de uma mesma categoria c_i associada a diferentes documentos, dentre os k documentos mais próximos de d_j. Nestes casos, $f(d_j, c_i)$ é o máximo de $\cos(\vec{d}_j, \vec{d}_m)$, para c_i pertencente a C_m e 1 <= m <= k.

A Tabela 2.1 mostra um exemplo de predição do categorizador k *NN* multi-rótulo com $k = 3$. A coluna 3-vizinhos mais próximos representa os documentos mais similares ao documento de teste d_j; a coluna Categorias pertinentes, as categorias associadas aos respectivos documentos mais similares; a coluna Valor do co-seno, o resultado da Equação (2.3) para os documentos mais similares; e a coluna Categorias preditas, o conjunto de categorias preditas pelo k *NN* multi-rótulo para o documento de teste d_j.

Tabela 2.1 - Exemplo de predição do k *NN* multi-rótulo para $k = 3$.

Documento teste	3 – vizinhos mais próximos	Categorias pertinentes	Valor do co-seno	Categorias preditas
d_j	d_1	$\{c_1, c_2\}$	0,8	$\{c_1$=0,8; c_2=0,8; c_3=0,4; c_4=0,3$\}$
	d_2	$\{c_2, c_3\}$	0,4	
	d_3	$\{c_1, c_3, c_4\}$	0,3	

2.5 Categorizador *ML-kNN*

O *Multi-Label k-Nearest Neighbor* (*ML-k NN*) é um categorizador multi-rótulo baseado no algoritmo k NN úni-rótulo [Zhang07]. Dado um documento de teste d_j, o *ML-k NN* identifica os k documentos da base de treinamento mais similares a d_j utilizando a métrica de distância co-seno (Equação (2.3)). Posteriormente, o algoritmo identifica a freqüência de cada categoria nestes k documentos. Utilizando esta informação, o *ML-k NN* prediz um conjunto de categorias para d_j utilizando o *maximum a posteriori principle* (MAP) [Sparacino00].

Formalmente, dado o documento $d_m \in TV$ e o conjunto de categorias pertinentes de d_m, $C_m \subseteq C$, podemos definir: (i) o vetor $\vec{y}_{d_m}(c_i)$, de tamanho igual a |C|, onde $\vec{y}_{d_m}(c_i)$ recebe 1 se $c_i \in C_m$ e zero caso contrário; e (ii) o conjunto dos k vizinhos mais próximos a d_m no conjunto de treinamento TV, $N(d_m)$.

Durante a fase de treinamento, baseado no conjunto de categorias associadas aos documentos d_m pertencentes a $N(d_m)$, um vetor de contagem de associação (*membership counting vector* [Zhang07]) de tamanho igual |C|, $\vec{C}_{d_m}(c_i)$, é computado segundo a Equação (2.4), abaixo, para cada $d_m \in TV$:

$$\vec{C}_{d_m}(c_i) = \sum_{a \in N(d_m)} \vec{y}_a(c_i) \tag{2.4}$$

O vetor $\vec{C}_{d_m}(c_i)$ sumariza a vizinhança de d_m em *TV* com respeito às categorias associadas aos documentos em $N(d_m)$.

Na fase de teste, para cada documento d_j em Te, o *ML-k NN* primeiramente identifica os k vizinhos mais próximos à d_j, $N(d_j)$, no conjunto TV. Seja $H_1^{c_i}$ um evento no qual a categoria c_i está associada a d_j; $H_0^{c_i}$ um evento no qual a categoria c_i não está associada a d_j; e $E_n^{c_i}(n \in \{0,1,\ldots,k\})$ um evento no qual existem exatamente n documentos associados à categoria c_i. Baseado no vetor de contagem de associação de d_j, $\vec{C}_{d_j}$, o vetor de categorias $\vec{y}_{d_j}$ pode ser determinado pelo *MAP*, conforme Equação (2.5).

$$\vec{y}_{d_j}(c_i) = \arg\max_{b \in \{0,1\}} P(H_b^{c_i} \mid E^{c_i}_{\vec{C}_{d_j}(c_i)}) \tag{2.5}$$

Pela regra de *Bayes*, a Equação **(2.5)** pode ser reescrita conforme Equação (2.6).

$$\vec{y}_{d_j}(c_i) = \arg\max_{b \in \{0,1\}} \frac{P(H_b^{c_i}) P(E^{c_i}_{\vec{C}_{d_j}(c_i)} \mid H_b^{c_i})}{P(E^{c_i}_{\vec{C}_{d_j}(c_i)})} \tag{2.6}$$

Eliminando o denominador $P(E^{c_i}_{\vec{C}_{d_j}(c_i)})$, pois é independente de $P(H^{c_i}_b)$, temos a equação final para a obtenção do vetor de categorias preditas para d_j:

$$\vec{y}_{d_j}(c_i) = \arg\max_{b\in\{0,1\}} P(H^{c_i}_b) P(E^{c_i}_{\vec{C}_{d_j}(c_i)} \mid H^{c_i}_b) \tag{2.7}$$

A Equação (2.7) mostra que, para determinar o vetor de categorias preditas $\vec{y}_{d_j}$, toda a informação sobre as probabilidades a priori, $P(H^{c_i}_b)$, e *a posteriori*, $P(E^{c_i}_{\vec{C}_{d_j}(c_i)} \mid H^{c_i}_b)$, são necessárias. Na verdade, essas probabilidades podem ser estimadas a partir da freqüência das categorias no conjunto de treinamento. A Figura 2.3 mostra o pseudocódigo do *ML-k NN* [Zhang07].

$[\vec{y}_{d_1}, \vec{r}_{d_1}] = ML-KNN(TV, k, d_j, s)$

%Computa a probabilidade a priori $P(H^{c_i}_\delta)$

(1) **para** $c_i \in C$ **faça**

(2) $P(H^{c_i}_1) = (s + \sum_{i=1}^{|TV|} \vec{y}_{x_i}(c_i)) / (sx2 + |TV|);\ \ P(H^{c_i}_0) = 1 - P(H^{c_i}_1);)$

%Computa a probabilidade a posterior $P(E^{c_i}_{\vec{C}_{d_1}(c_i)} \mid H^{c_i}_\delta)$

(3) **Identifica** $N(x_i),\ i \in \{1,2,\ldots,|TV|\};$

(4) **para** $c_i \in C$ **faça**

(5) **para** $j \in \{0,1,\ldots,k\}$ **faça**

(6) $c[j] = 0;\ \ c'[j] = 0;$

(7) **para** $l \in \{0,1,2,\ldots,|TV|\}$ **faça**

(8) $\delta = \vec{C}_{x_l}(c_i) = \sum_{a\in N(x_l)} \vec{y}_a(c_i);$

(9) **se** $(\vec{y}_{x_l}(c_i) == 1)$ **então** $c[\delta] = c[\delta] + 1;$

(10) **senão** $c'[\delta] = c'[\delta] + 1;$

(11) **para** $j \in \{0,1,\ldots,k\}$ **faça**

(12) $P(E^{c_i}_j \mid H^{c_i}_1) = (s + c[j]) / (sx(k+1) + \sum_{p=0}^{k} c[p]);$

(13) $P(E^{c_i}_j \mid H^{c_i}_0) = (s + c'[j]) / (sx(k+1) + \sum_{p=0}^{k} c'[p]);$

%Computa $\vec{y}_{d_1}$ e $\vec{r}_{d_1}$

(14) **Identifica** $N(d_j);$

(15) **para** $c_i \in C$ **faça**

(16) $\vec{C}_{d_1}(c_i) = \sum_{a\in N(d_1)} \vec{y}_a(c_i);$

(17) $\vec{y}_{d_1}(c_i) = \arg\max_{\delta\in\{0,1\}} P(H^{c_i}_\delta) P(E^{c_i}_{\vec{C}_{d_1}(c_i)} \mid H^{c_i}_\delta);$

(18) $\vec{r}_{d_1}(c_i) = P(H^{c_i}_1 \mid E^{c_i}_{\vec{C}_{d_1}(c_i)}) = (P(H^{c_i}_1) P(E^{c_i}_{\vec{C}_{d_1}(c_i)} \mid H^{c_i}_1)) / P(E^{c_i}_{\vec{C}_{d_1}(c_i)});$

Figura 2.3 - Pseudocódigo do algoritmo *ML-k NN*.

Os parâmetros de entrada do algoritmo são TV, k, d_j e s. O parâmetro s controla a suavização da probabilidade a priori, e neste trabalho, optamos em utilizar o valor $s=1$ (suavização Laplaciana [Zhang07]). De acordo com a Figura 2.3, os passos (1) e (2) calculam a probabilidade a priori, $P(H_b^{c_i})$. Os passos de (3) a (13) estimam a probabilidade a *posteriori*, $P(E^{c_i}_{\vec{C}_{d_j}(c_i)} \mid H_b^{c_i})$, onde $c[j]$ contabiliza o número de documentos entre os k documentos similares no conjunto de treinamento que possuem a categoria c_i. Correspondentemente, $c'[j]$ contabiliza o número de documentos entre os k documentos similares no conjunto de treinamento que não possuem a categoria c_i. Finalmente, os passos (14) a (18) são a predição do algoritmo, isto é, a atribuição de um grau de crença para cada categoria $c_i \subseteq C$ referente ao documento de teste d_j, $f(.,c_i)$.

2.6 Categorizador VG-RAM

Uma rede neural artificial (*Artifical Neural Network – ANN*) é um modelo de computação inspirado na forma como a estrutura paralela e densamente conectada do cérebro dos mamíferos processa as informações. Mais formalmente, as redes neurais artificiais são sistemas paralelos distribuídos compostos por unidades de processamento simples, chamados de nós, que calculam determinadas funções matemáticas (normalmente não-lineares) [Haykin99]. Essas unidades são dispostas em uma ou mais camadas e interligadas por um número de conexões, chamadas de sinapses.

Essencialmente, um neurônio artificial é composto por um conjunto de sinapses, um somador e uma função de transferência (ou função de ativação) [Haykin99]. Conforme a Figura 2.4, cada sinapse do neurônio k está associada aos pesos $\{w_{k1}, w_{k2}, \ldots w_{km}\}$. Especificamente, ao ser apresentada uma informação ao neurônio, $\{x_1, x_2, \ldots x_m\}$, cada elemento da informação é multiplicado pelo peso w_{kj} da sinapse, e o resultado de cada entrada é somado, ou seja, é realizada uma soma ponderada da informação de entrada pelo Somador. O resultado da soma passa por uma Função de ativação, $\varphi(.)$, que computa a saída y_k do neurônio em função da saída do Somador.

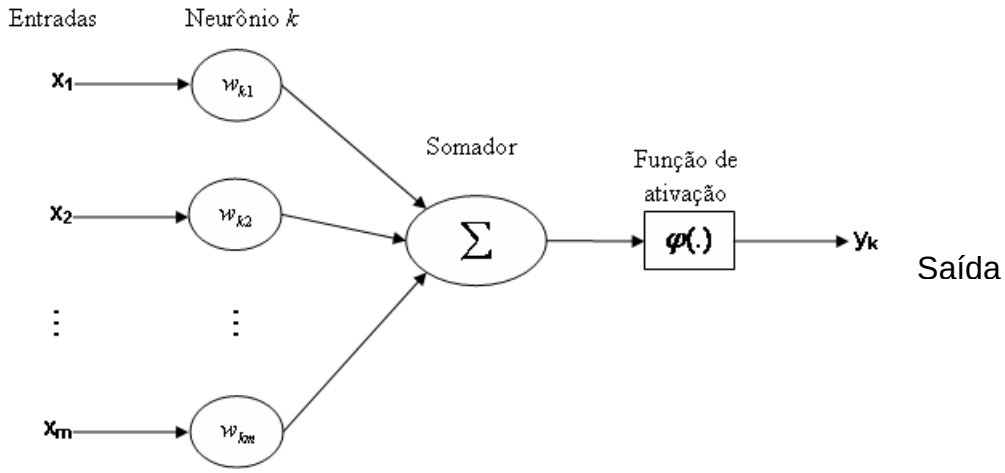


Figura 2.4 - Esquema de um neurônio artificial.

Redes neurais sem peso (RNSP), também conhecidas como redes neurais baseadas em *Random Access Memories* (RAM), não armazenam conhecimento em suas conexões, mas em memórias do tipo RAM dentro dos nodos da rede, ou neurônios. Estes neurônios operam com valores de entrada binários e usam RAM como tabelas-verdade: as sinapses de cada neurônio coletam um vetor de bits da entrada da rede, que é usado como o endereço da RAM, e o valor armazenado neste endereço é a saída do neurônio. O treinamento pode ser feito em um único passo e consiste basicamente em armazenar a saída desejada no endereço associado com o vetor de entrada do neurônio.

Apesar da sua notável simplicidade, as RNSP são muito efetivas como ferramentas de reconhecimento de padrões, oferecendo treinamento e teste rápidos, e fácil implementação. No entanto, se a entrada da rede for muito grande, o tamanho da memória dos neurônios da RNSP torna-se proibitivo, dado que tem de ser igual a 2^n, onde n é o tamanho da entrada. As redes *Virtual Generalizing RAM* (*VG-RAM*) são redes neurais baseadas em RAM que somente requerem capacidade de memória para armazenar os dados relacionados ao conjunto de treinamento.

Os neurônios *VG-RAM* armazenam os pares entrada-saída observados durante o treinamento, em vez de apenas a saída. Na fase de teste, as memórias dos neurônios *VG-RAM* são pesquisadas mediante a comparação entre a entrada apresentada à rede e todas as entradas nos pares entrada-saída aprendidos. A saída de cada neurônio *VG-RAM* é determinada pela saída do par cuja entrada é a mais próxima da entrada apresentada – a função de distância adotada pelos neurônios *VG-RAM* é a distância de *Hamming*, isto é., o número de bits diferentes entre dois vetores de bits de igual tamanho. Se existir mais do que um par na

mesma distância mínima da entrada apresentada, a saída do neurônio é escolhida aleatoriamente entre esses pares.

Neste trabalho utilizamos duas implementações de rede neural *VG-RAM* para avaliar o impacto de empates no desempenho do categorizador em problemas de categorização de atividades econômicas. A primeira, *VG-RAM WNN* [Badue08, DeSouza07, DeSouza09a], transforma um problema multi-rótulo em n problemas independentes úni-rótulo, onde n é o número de possíveis categorias de cada documento. A segunda, *VG-RAM WNN-COR* [DeSouza08, DeSouza09b], explora a correlação das categorias associadas a cada documento. Essas duas implementações são apresentas nas subseções seguintes.

2.6.1 VG-RAM WNN

A Tabela 2.2 ilustra a tabela-verdade de um neurônio *VG-RAM* com três sinapses (X_1, X_2 e X_3). Esta tabela-verdade contém três pares entrada-saída que foram armazenados durante a fase de treinamento ($par\ \#1$, $par\ \#2$ e $par\ \#3$). Durante a fase de teste, quando um vetor de entrada é apresentado à rede, o algoritmo de teste *VG-RAM* calcula a distância entre este vetor de entrada e cada entrada dos pares entrada-saída armazenados na tabela-verdade. No exemplo da Tabela 2.2, a distância de *Hamming* entre o vetor de entrada (*input*) e o $par\ \#1$ é dois, porque ambos os bits X_2 e X_3 não são semelhantes aos bits X_2 e X_3 do vetor de entrada. A distância do $par\ \#2$ é um, porque X_1 é o único bit diferente. A distância do $par\ \#3$ é três. Portanto, para este vetor de entrada, o algoritmo avalia a saída do neurônio, Y, como "$categoria\ 2$", pois é o valor de saída armazenado no $par\ \#2$.

Tabela 2.2 - Exemplo de tabela-verdade de um neurônio da RNSP *VG-RAM WNN* [SCAE08].

Tabela-verdade	X_1	X_2	X_3	Y
$par\ \#1$	1	1	0	$categoria\ 1$
$par\ \#2$	0	0	1	$categoria\ 2$
$par\ \#3$	0	1	0	$categoria\ 3$
	↑	↑	↑	↓
vetor de entrada	1	0	1	$categoria\ 2$

Para categorizar documentos de texto usando uma RNSP *VG-RAM*, um documento é representado por um vetor multidimensional $V = \{v_1, v_2, \ldots, v_{|V|}\}$, onde cada elemento v_i

corresponde a um peso associado a um termo específico do vocabulário de interesse. Uma RNSP *VG-RAM* de uma única camada (Figura 2.5) é utilizada, de forma que as sinapses $X = \{x_1, x_2, \ldots, x_{|X|}\}$ de seus neurônios são conectadas aleatoriamente à entrada da rede $N = \{n_1, n_2, \ldots, n_{|N|}\}$, que tem o mesmo tamanho de um vetor que representa um documento, isto é, $|N| = |V|$. Note que $|X| < |V|$ (nossos experimentos demonstraram que $|X| < |V|$ provê melhor desempenho). Cada sinapse x_i de um neurônio forma uma célula *Minchinton* com a próxima x_{i+1} ($x_{|X|}$ forma uma célula *Minchinton* com x_1) [Mitchell98]. O tipo de célula *Minchinton* usada retorna 1 se a sinapse x_i da célula é conectada a um elemento de entrada n_j cujo valor é maior do que aquele do elemento n_k ao qual a sinapse x_{i+1} é conectada (isto é, $n_j > n_k$); caso contrário, ela retorna zero.

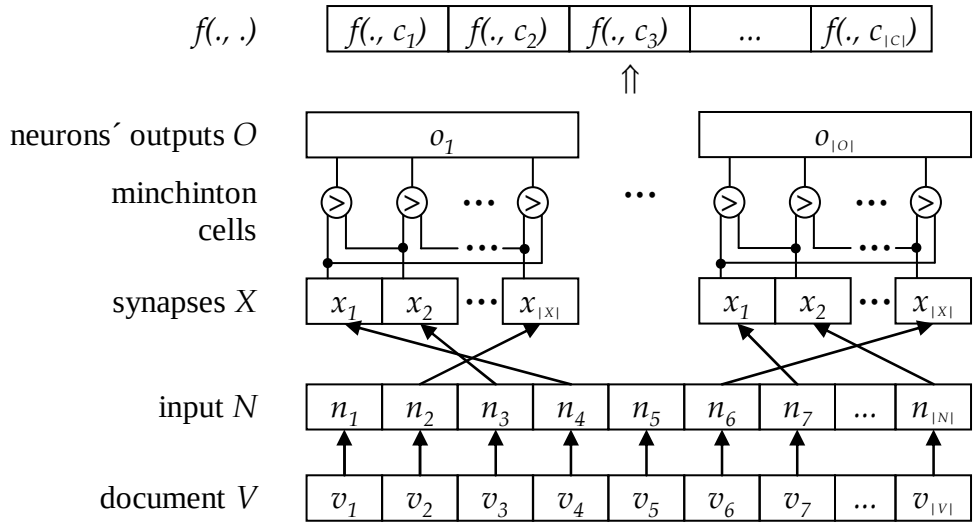


Figura 2.5 – Arquitetura para categorização de texto da RNSP *VG-RAM WNN* [SCAE08].

Durante a fase de treinamento, para cada documento no conjunto de treinamento, o vetor correspondente V é conectado à entrada N da RNSP *VG-RAM* e as saídas $O = \{o_1, o_2, \ldots, o_{|O|}\}$ dos neurônios a uma das categorias do documento. Todos os neurônios da RNSP *VG-RAM* são então treinados para retornar como saída esta categoria com este vetor de entrada. O treinamento para este vetor de entrada é repetido para cada categoria associada ao documento correspondente. Durante a fase de teste, para cada documento de teste, a entrada é conectada ao vetor correspondente e o número de neurônios retornado para cada categoria é contabilizado. A saída da rede é computada dividindo-se a contagem de cada categoria pelo número de neurônios da rede.

A saída da rede é reorganizada como um vetor cujo tamanho é igual ao número de categorias existentes. O valor de cada elemento deste vetor varia entre 0 e 1 e representa a porcentagem de neurônios que exibiram a categoria correspondente como saída (a soma dos valores de todos os elementos deste vetor é sempre 1). Desta forma, a saída da rede reorganizada deste modo implementa a função $f(.,.)$, que apresenta valores no domínio dos números reais e que mapeia a múltipla pertinência de um documento frente a um dado conjunto de categorias existentes. Finalmente, um valor limiar τ_i para cada categoria c_i pode ser usado com a função $f(.,.)$, a fim de definir o conjunto de categorias a serem atribuídas a um documento de teste d_j: se $f(d_j, c_i) \geq \tau_i$, então c_i é atribuída a d_j.

2.6.2 VG-RAM WNN-COR

Enquanto numa RNSP *VG-RAM* cada neurônio é treinado para retornar como saída uma única categoria para cada vetor de entrada, numa RNSP *VG-RAM* com Correlação de Dados (RNSP *VG-RAM WNN-COR* [DeSouza08, DeSouza09b]) cada neurônio pode ser treinado para retornar como saída um conjunto de categorias para cada vetor de entrada. A Tabela 2.3 ilustra a tabela-verdade de uma RNSP *VG-RAM WNN-COR* com três sinapses X_1, X_2 e X_3 e três pares entrada-saída armazenados durante a fase de treinamento (*par* #1, *par* #2 e *par* #3). Semelhante à RNSP *VG-RAM*, quando um vetor de entrada é apresentado à rede na fase de teste, o algoritmo de teste da RNSP *VG-RAM WNN-COR* computa a distância entre este vetor de entrada e cada entrada dos pares entrada-saída na tabela-verdade. No exemplo da Tabela 2.3, a distância de *Hamming* entre o vetor de entrada (*input*) e os pares #1, #2, e #3 é dois, um e três, respectivamente. Como o *par* #2 da tabela-verdade é o mais próximo da entrada da rede, a saída do neurônio da RNSP *VG-RAM WNN-COR* é dada pelas categorias 1 e 3, isto é, o valor de Y representa ambas as categorias, 1 e 3.

Tabela 2.3 - Exemplo de tabela-verdade de uma rede neural *VG-RAM WNN-COR* [SCAE08].

Tabela-verdade	X_1	X_2	X_3	Y
par #1	1	1	0	*categoria* 1
par #2	0	0	1	*categoria* 1; *categoria* 3
par #3	0	1	0	*categoria* 1; *categoria* 2; *categoria* 3
	↑	↑	↑	↓

vetor de entrada	1	0	1	*categoria* 1; *categoria* 3

Para categorizar documentos de texto usando uma RNSP *VG-RAM WNN-COR*, a mesma configuração da RNSP *VG-RAM*, ilustrada na Tabela 2.3, é usada. Na fase de treinamento, para cada documento no conjunto de treinamento, o vetor correspondente V é conectado à entrada da RNSP *VG-RAM WNN-COR*, N, e as saídas dos seus neurônios, O, ao conjunto de categorias atribuído ao documento. Cada neurônio da RNSP *VG-RAM WNN-COR* é treinado para retornar como saída este conjunto com este vetor de entrada. Durante a fase de teste, para cada documento de teste, o vetor correspondente V é conectado à entrada da rede, N. A função $f(.,.)$ é computada ao dividir o número de votos para cada categoria pelo número total de categorias retornadas pela rede. O número de votos para cada categoria é obtido ao contar suas ocorrências em todos os conjuntos retornados pelos neurônios da rede.

2.7 Aplicação de Categorização Multi-rótulo de Texto

Devido ao aumento da disponibilidade do número de documentos de texto no formato digital, e pela conseqüente necessidade de organizá-los, a Categorização de Texto (CT) tornou-se uma das técnicas chave para manipular e organizar dados no formato texto. Hoje em dia, a CT pode ser aplicada em diversos problemas, tal como: organização de documentos, filtragem de texto, geração automatizada de metadados, desambiguação do sentido da palavra, categorização de páginas Web baseados em um catálogo hierárquico [Sebastiani02], entre outras. No entanto, existem muitas outras importantes aplicações às quais pouca atenção tem sido dada. Um exemplo é a categorização de atividades econômicas baseada na descrição dos propósitos de uma empresa, ou seja, as atividades realizadas por uma empresa [Badue08, Ciarelli08, Ciarelli09, DeSouza07, DeSouza08, DeSouza09a, DeSouza09b, Oliveira08a, Oliveira08b]. Neste trabalho, verificamos o impacto do tratamento de empate no desempenho dos categorizadores multi-rótulo de texto utilizando bases de texto contendo descrições de atividades econômicas de empresas brasileiras.

2.7.1 Categorização de atividades econômicas

A categorização de companhias de acordo com as respectivas atividades exercidas é uma etapa importante do processo de obtenção de informações para a realização de análises estatísticas das atividades econômicas de uma cidade ou país. Com as companhias categorizadas, é possível realizar uma análise estruturada de cada setor da economia, auxiliando empresas e governos em suas decisões.

Para facilitar e melhorar a qualidade de categorização das empresas de acordo com as atividades econômicas, o governo brasileiro está criando uma biblioteca digital centralizada com as declarações de propósitos de todas as empresas no país. Esta biblioteca vai ajudar as três esferas de governo – federal, os 27 Estados, e os mais de 5.000 municípios brasileiros – na tarefa de categorizar as empresas de acordo com a lei Brasileira vigente.

A categorização oficial das atividades econômicas adotada pelos órgãos da administração federal é baseada na Classificação Nacional de Atividades Econômicas (CNAE). A CNAE foi desenvolvida tendo como referência a *International Standard Industrial Classification of All Economic Activities - ISIC*, 3ª revisão, das Nações Unidas. A *ISIC* é uma padronização internacional definida pelas Nações Unidas para a disseminação das estatísticas econômicas no mundo. A partir da elaboração da CNAE foi derivada outra classificação, a CNAE-FISCAL, ou CNAE-Subclasses [CNAE03], que é um detalhamento das Classes da CNAE para uso nos cadastros da administração pública, em especial da administração tributária, nas três esferas do governo. A Tabela 2.4 apresenta sumariamente a CNAE-Subclasses Versão 1.1.

Tabela 2.4 – Apresentação sumária da Tabela CNAE-Subclasses, Versão 1.1.

Seções	Divisões	Grupos	Classes	Subclasses	Denominação
A	2	7	25	91	Agricultura, pecuária, silvicultura e exploração florestal
B	1	1	2	11	Pesca
C	4	7	14	42	Indústrias extrativas
D	23	104	286	395	Indústrias de transformação
E	2	4	7	8	Produção e distribuição de eletricidade, gás e água
F	1	6	16	43	Construção
G	3	19	72	223	Comércio; Reparação de veículos automotores, objetos pessoais e domésticos
H	1	2	7	16	Alojamento e alimentação
I	5	14	29	76	Transporte, armazenagem e comunicações
J	3	11	27	65	Intermediação financeira, seguros, previdência complementar e serviços relacionados
K	5	24	38	80	Atividades imobiliárias, aluguéis e serviços prestados às empresas
L	1	3	10	10	Administração pública, defesa e seguridade social

M	1	4	10	17	Educação
N	1	3	9	35	Saúde e serviços sociais
O	4	11	26	69	Outros serviços coletivos, sociais e pessoais
P	1	1	1	1	Serviços domésticos
Q	1	1	1	1	Organismos internacionais e outras instituições extraterritoriais
Total	**59**	**222**	**580**	**1.183**	

A CNAE-Subclasses é uma tabela hierárquica de descrição de atividades econômicas com os respectivos códigos associados. Conforme a Tabela 2.4 mostra, a CNAE-Subclasses 1.1 está organizada hierarquicamente em 5 níveis: Seção, Divisão, Grupo, Classe e Subclasse, contendo 17 Seções, 59 Divisões, 222 Grupos, 580 Classes e 1.183 Subclasses. O campo Denominação representa a descrição textual do código de Seção. Cada código nos níveis Divisão, Grupo, Classe e Subclasse também estão associados a uma denominação [CNAE03].

Os códigos da CNAE-Subclasses são constituídos por 7 dígitos, sendo os 5 primeiros dígitos referentes ao nível de Classe e os dois últimos referente ao detalhamento de cada Classe CNAE. Por exemplo, a Figura 2.6 apresenta o nível de Subclasse da Seção A para o código 0111-2/01 com a denominação “CULTIVO DE ARROZ”. Como podemos perceber, a Classe é identificada pelo código 0111-2 e pela denominação “CULTIVO DE CEREAIS PARA GRAOS”.

Ministério do Planejamento, Orçamento e Gestão

CONCLA
Comissão Nacional de Classificação

Topo da Estrutura... | Nova Pesquisa...

CNAE-FISCAL 1.1

Hierarquia		
Seção:	A	AGRICULTURA, PECUARIA, SILVICULTURA E EXPLORAÇAO FLORESTAL
Divisão:	01	AGRICULTURA, PECUARIA E SERVIÇOS RELACIONADOS
Grupo:	011	PRODUÇAO DE LAVOURAS TEMPORARIAS
Classe:	0111-2	CULTIVO DE CEREAIS PARA GRAOS
Subclasse	**0111-2/01**	**CULTIVO DE ARROZ**

Lista de Atividades...

Notas Explicativas:

Esta subclasse compreende:
O cultivo de arroz

Esta subclasse compreende também:
O beneficiamento do arroz em estabelecimento agrícola, quando complementar ao cultivo
A produçao de semente para plantio de arroz, quando complementar ao cultivo

Esta subclasse não compreende:
O beneficiamento do arroz, em estabelecimento nao agrícola (1551-2/01)
O beneficiamento do arroz, realizado por terceiros, em estabelecimento agrícola (0161-9/05)
A produçao de óleo de arroz em bruto (1531-8/00)
O serviço de colheita realizado por terceiros (0161-9/04)
A produçao de sementes certificadas de arroz (0119-8/17)
O serviço de preparaçao de terreno de cultivo realizado por terceiros (0161-9/99)

Figura 2.6 – Um exemplo da tabela CNAE para o nível de Subclasse.

Os códigos Subclasse também carregam a identificação dos níveis de Divisão e Grupo. Por exemplo, para o código 0111-2/01, Figura 2.6, os dois primeiros dígitos, 01, identificam o nível de Divisão, com a denominação “AGRICULTURA, PECUARIA E SERVIÇOS RELACIONADOS”, e os três primeiros, 011, o de Grupo, com a denominação “PRODUÇAO DE LAVOURAS TEMPORARIAS”.

Além da denominação do código de um determinado nível, existem notas explicativas para agregar mais informação àquele nível. No caso do nível de Subclasse, as notas explicativas mostram o que a Subclasse compreende (“Esta subclasse compreende:”), o que ela compreende também (“Esta subclasse compreende também:”) e o que ela não compreende (“Esta subclasse não compreende:”).

Atualmente, em muitos órgãos usuários a determinação de quais códigos devem ser atribuídos a cada empresa, a codificação em CNAE-Subclasses, é feita manualmente por codificadores humanos treinados para tal e apoiados por ferramentas computacionais de busca em versões eletrônicas da tabela CNAE-Subclasses. O codificador (ou categorizador) humano treinado deve associar/combinar a descrição da atividade da empresa com a informação na tabela CNAE-Subclasses e com seu conhecimento, fruto de seus vários anos de educação e experiência profissional, para atribuir códigos CNAE-Subclasse.

Conforme as características apresentadas anteriormente, o problema de categorização de atividades econômicas consiste em, dada uma descrição textual do propósito de uma empresa, categorizá-la em um ou mais dos 1.183 possíveis códigos (ou categorias) CNAE-Subclasse. O grande número de possíveis categorias torna este problema complexo quando comparado com outros apresentados na literatura [Sebastiani02]. Em particular, o grande número de categorias torna os empates na saída dos categorizadores mais prováveis. Por essa razão escolhemos este problema de categorização para este trabalho.

3 METODOLOGIA

Como discutido no Capítulo 2, existem diversas técnicas para categorização multi-rótulo de texto, mas, aparentemente, não existe uma única que apresente sempre o melhor desempenho para todos os problemas [Monard03]. Desta forma, é importante compreender a limitação das diferentes técnicas utilizando alguma metodologia de avaliação que permita compará-las.

Neste capítulo descrevemos uma metodologia de avaliação, representada graficamente na Figura 3.1, que permite analisar experimentalmente o impacto dos tipos de *ranking* Ordinal Aleatório, Denso, Padrão e Modificado sobre as métricas de avaliação de desempenho multi-rótulo utilizando os categorizadores apresentados no Capítulo 2. Esta metodologia foi elaborada com base no fluxo de execução de experimentos da ferramenta SCAE [SCAE08].

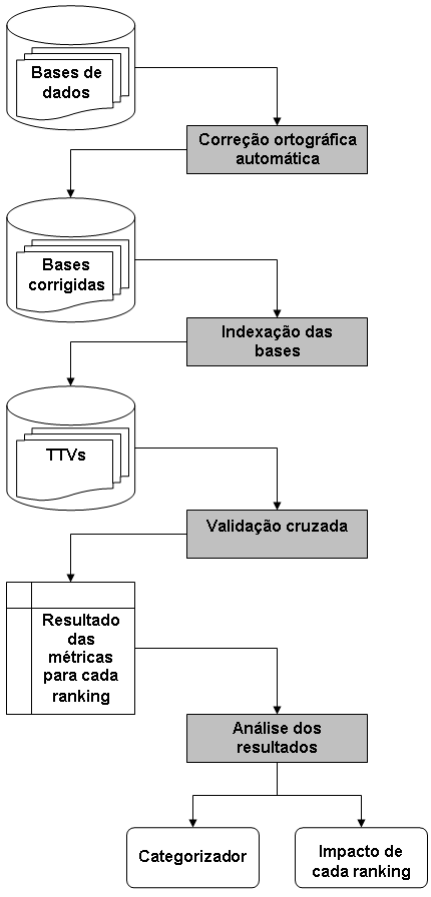


Figura 3.1 – Fluxograma da metodologia de avaliação dos categorizadores e do impacto de cada *ranking*.

Na Figura 3.1, os retângulos na cor cinza (objetos mais à direita na figura) representam os procedimentos que são realizados nos dados de entrada/saída (objetos mais à esquerda). Conforme a Figura 3.1 mostra, definidas as *Bases de dados* para a realização dos experimentos, realizamos a *Correção ortográfica automática* das mesmas. Com as *Bases corrigidas*, o procedimento de *Indexação das bases* é realizado, que transforma documentos textuais em vetores de pesos (ver Seção 2.3, pág. 26), que chamamos aqui de *Train and Test Vectors* (*TTVs*). Esses vetores são utilizados para realizar o treinamento (e validação) e o teste dos categorizadores. Empregamos 10-*fold cross-validation*, representada na figura pelo procedimento *Validação cruzada*, para tornar possível o teste estatístico de hipóteses relacionadas ao impacto dos diferentes tipos de *rankings* nas métricas. Esse procedimento tem como saída 10 resultados de desempenho de um determinado categorizador segundo cada uma das métricas de avaliação para os *rankings* Ordinal Aleatório, Denso, Padrão e Modificado. Diante desses resultados, por meio do teste estatístico t de *Student*, analisamos o impacto dos tipos de *ranking* estudados no desempenho dos categorizadores k *NN*, *ML-*k *NN*, *VG-RAM WNN e VG-RAM WNN-COR* segundo as métricas de avaliação de desempenho multi-rótulo mais populares da literatura. Cada um dos procedimentos mostrados na Figura 3.1 é descrito em maiores detalhes nas seções seguintes deste capítulo.

Todos esses procedimentos, com exceção de *Análise dos resultados*, estão definidos/implementados na ferramenta SCAE. Instalamos esta ferramenta no cluster *Enterprise3* do Laboratório de Computação de Alto Desempenho (LCAD) para realizar todos os experimentos deste trabalho. Esta ferramenta é um ambiente de desenvolvimento com contínuas melhorias que chamamos de revisões. Neste trabalho, utilizamos a revisão 742 para a realização dos nossos experimentos.

3.1 Bases de dados

O conjunto de dados empregado em nossa avaliação experimental é composto de descrições textuais de atividades econômicas de empresas brasileiras. Todas essas descrições foram manualmente categorizadas em uma ou mais atividades econômicas por funcionários públicos Brasileiros treinados nesta tarefa. A lei brasileira determina que todas as empresas devem apresentar uma descrição textual das suas atividades econômicas para órgãos do governo para que elas sejam categorizadas de acordo com a tabela oficial de atividades

econômicas, Tabela CNAE-Subclasse [CNAE03]. Chamamos de documento a descrição textual das atividades econômicas de uma empresa categorizadas em uma ou mais categorias da tabela CNAE-Sublcasses.

Neste trabalho, contamos com descrições de atividades econômicas de empresas das cidades de Vitória – Espírito Santo e Belo Horizonte – Minas Gerais. A base de dados de Vitória, chamada de VIX, possui 3.281 documentos referentes a empresas da localidade categorizados em 764 diferentes categorias CNAE-Subclasse. O número médio de categorias por documento é 4,3 (desvio padrão de 5,6).

A Figura 3.2 apresenta o histograma do número de documentos com um determinado número de categorias. No gráfico da Figura 3.2, o eixo horizontal representa o Número de categorias por documento e o eixo vertical o Número de documentos. De 1 a 35 categorias por documento, as barras do gráfico indicam exatamente o número de documentos com o respectivo número de categorias. De 36 categorias por documento em diante, só aparecem no eixo horizontal do gráfico os números de categorias por documento para os quais há documentos na base VIX.

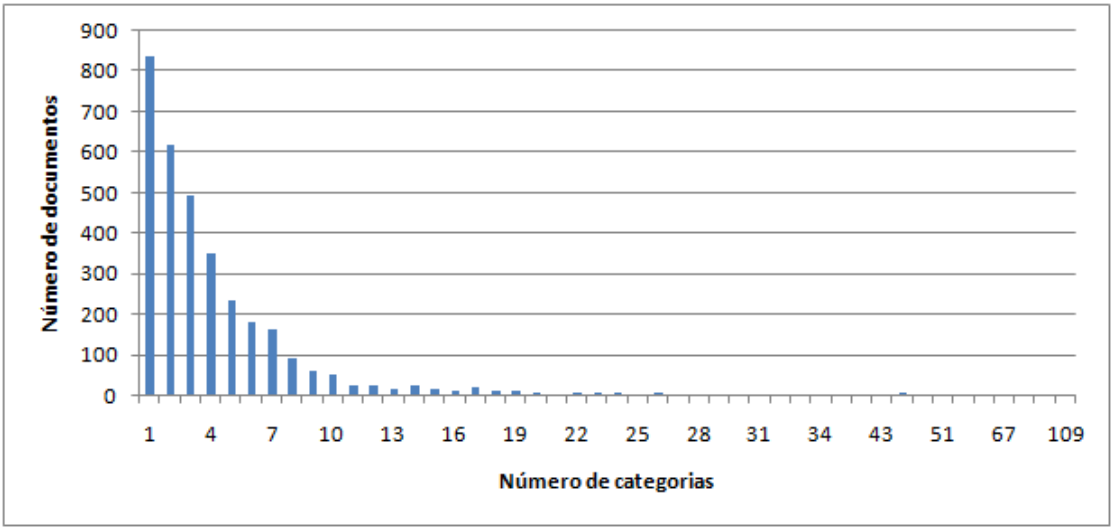


Figura 3.2 – Distribuição do número de categorias por documento na base de dados VIX.

O número de categorias por documento varia de 1 a 109, sendo que mais de 800 documentos possuem apenas uma categoria e apenas um documento possui 109 categorias. Como a Figura 3.2 mostra, a maior parte dos documentos da base VIX possui de 1 a 7 categorias por documento (87,53%).

A base de dados de Belo Horizonte, chamada BH, possui 88.000 documentos categorizados em 1.002 diferentes categorias CNAE-Subclasse. O número médio categorias

por documento é 2,0 (desvio padrão de 1,7). A Figura 3.3 apresenta o histograma da base BH.

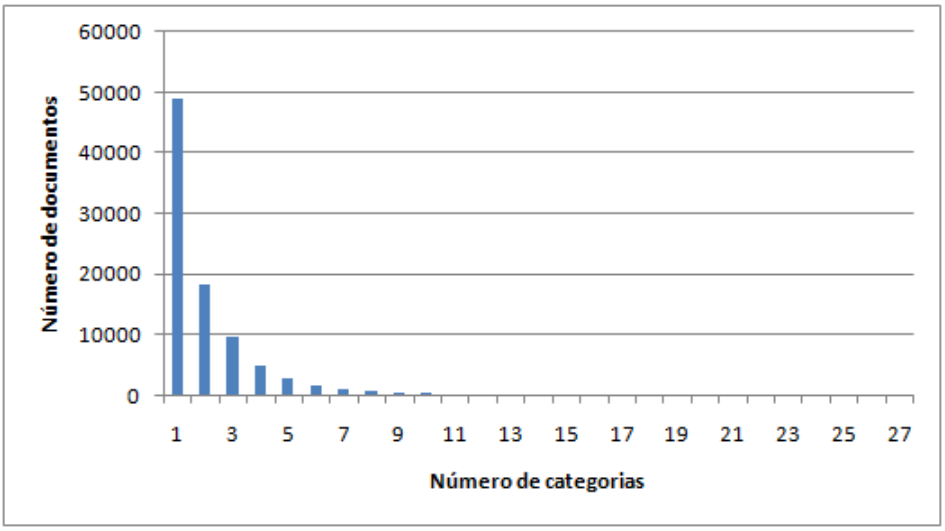


Figura 3.3 – Distribuição do número de categorias por documento na base de dados BH.

Na base BH, o número de categorias por documento varia entre 1 e 27, sendo que quase 50000 documentos possuem apenas uma categoria e apenas um documento possui 27 categorias. Como a Figura 3.3 mostra, a maior parte dos documentos da base BH possui entre 1 e 3 categorias.

A partir das bases VIX e BH, geramos duas bases de dados que utilizamos para treinar, validar, testar e avaliar o impacto dos tipos de *ranking* nos categorizadores. A primeira base gerada, chamada de EX100 (EXatamente 100), possui exatamente 100 exemplares de documentos de cada categoria. Ela é composta de 6.911 documentos selecionados aleatoriamente da união de VIX e BH; 105 categorias diferentes ocorrem na base EX100, isto é, existem exatamente 100 documentos na base categorizados dentro de cada uma destas 105 categorias. O número médio de categorias por documento é 1,52 (desvio padrão de 0,79).

As características da EX100 permitem avaliar o impacto dos *rankings* Ordinal Aleatório, Denso, Padrão e Modificado no desempenho dos categorizadores nos casos onde as categorias estão aproximadamente uniformemente distribuídas na base de treinamento. A Figura 3.4 apresenta o histograma da base EX100. Conforme a figura mostra, o número de categorias por documento varia de 1 a 6, sendo que mais de 4.000 documentos possuem apenas uma categoria e 9 documentos possuem 6 categorias. A maior parte dos documentos desta base possui entre 1 e 2 categorias (89,22%).

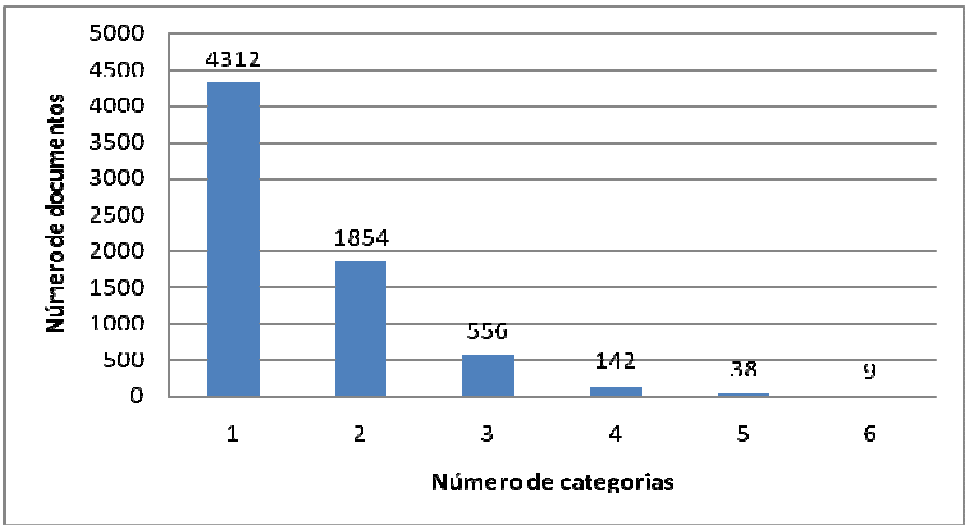


Figura 3.4 – Distribuição do número de categorias por documento na base de dados EX100.

Na segunda base gerada, chamada de AT100 (ATé 100), cada categoria ocorre em até 100 diferentes documentos, isto é, existem entre 1 e 100 exemplares de documentos de cada categoria. Ela é composta de 10.495 documentos selecionados aleatoriamente da união de VIX e BH; 692 categorias diferentes ocorrem na base AT100. O número médio de categorias por documentos é 1,49 (desvio padrão de 0,86). As características de AT100 permitem avaliar o impacto de cada tipo de *ranking* no desempenho dos categorizadores nos casos onde existem categorias raras.

A Figura 3.5 apresenta o histograma da base AT100. Conforme a figura mostra, o número de categorias por documento varia de 1 a 12, sendo que mais de 7.000 documentos possuem apenas uma categoria e um documento possui 12 categorias. A maior parte dos documentos desta base possui entre 1 e 2 categorias.

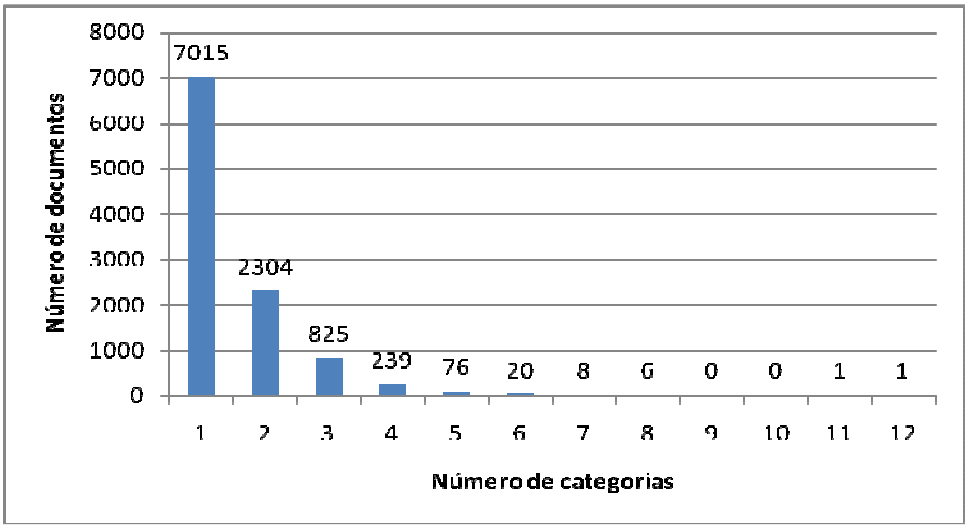


Figura 3.5 – Distribuição do número de categorias por documento na base de dados AT100.

Além das bases EX100 e AT100, utilizamos a própria tabela CNAE-Subclasse, chamada de CNAE, para treinar os categorizadores. A tabela CNAE-Subclasse possui 1183 Subclasses. Cada uma destas Subclasses possui um pequeno texto com sua denominação (ver Seção 2.7.1, pág. 39). Este texto foi utilizado, juntamente com o código CNAE correspondente, como documento de treinamento. Foram utilizados apenas os documentos cujas categorias ocorrem nas bases EX100 ou AT100. Então, temos duas bases CNAE: uma para a base EX100, chamada CNAE_EX100, com 105 documentos (códigos CNAE-Subclasse), e outra para a AT100, chamada CNAE_AT100, com 692 documentos. Estas bases foram usadas porque, no caso de problema de categorização em CNAE, esta informação estará sempre disponível e verificamos que utilizá-la melhora o desempenho dos categorizadores.

3.2 Correção ortográfica automática

Antes da geração das bases (EX100 e AT100) para os experimentos de 10-*fold cross-validation*, realizamos o procedimento de correção ortográfica automática das bases VIX, BH e CNAE (Figura 3.1). Foi adotada a correção automática ao invés da manual em função do grande número de documentos existentes nas bases.

A correção ortográfica está relacionada a dois principais problemas: a detecção de erro, que é o processo de encontrar uma palavra errada; e a correção de erro, que é o processo de sugerir palavras corretas para substituir uma palavra errada encontrada [Martins04]. Atualmente, existem corretores ortográficos para diversos idiomas. Dentre os existentes para o Português escolhemos o *GNU Aspell* [Aspell08] por ter código aberto e, assim, permitir a customização necessária para seu uso no SCAE [SCAE08].

A ferramenta *Aspell* faz uso de um dicionário para propor uma lista de palavras corretas para uma palavra errada. Basicamente, a ferramenta calcula a distância entre a palavra errada e cada uma das palavras existentes no dicionário, sendo que a de menor distância é colocada no topo da lista de sugestões, ou seja, a topo da lista é a considerada correta. O valor da distância é considerado pelo *Aspell* como uma pontuação (*score*).

Em testes preliminares de correção ortográfica automática, percebemos que em muitas situações a palavra correta estava na lista de sugestões do *Aspell*, mas não se encontrava no topo. Visando melhorar o desempenho, utilizamos uma lista auxiliar de palavras com as

respectivas freqüências [Crowell03]. Esta lista foi gerada a partir das palavras existentes nos documentos da base VIX corrigida manualmente.

O novo *score*, que chamamos de *rank*, é calculado a partir do *score* atribuído pelo *Aspell* e a freqüência da palavra (FP) existente na lista auxiliar, conforme Equação (3.1). Então, para que o *Aspell* retorne uma palavra correta dada uma errada, o mesmo escolhe a de menor *rank*.

$$rank = \frac{score}{1 + \ln(FP)} \tag{3.1}$$

Mais detalhes sobre o corretor ortográfico automático empregado em [SCAE08].

3.3 Indexação das bases de dados

O procedimento de indexação é realizado após o pré-processamento das bases corrigidas, que envolve [Sebastiani02]: *Análise léxica*; *Remoção de stopwords* (artigos, preposições, etc.); e *Redução de dimensionalidade*. A Figura 3.6 apresenta graficamente o fluxograma do pré-processamento, que está também definido/implementado na ferramenta SCAE.

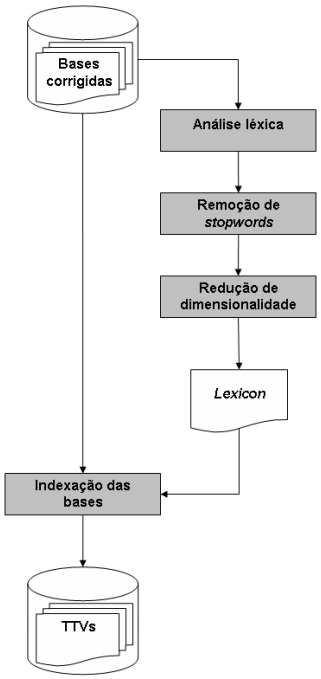


Figura 3.6 – Fluxograma do pré-processamento realizado nas Bases corrigidas anterior à indexação.

Na *Análise léxica*, os textos dos documentos são convertidos em um conjunto de palavras, que são candidatas a serem adotadas como termos dos documentos. Para isso, as palavras do texto dos documentos são separadas pelos caracteres de espaço e pontuação, ou seja, esses caracteres são delimitadores das palavras dos documentos. Por exemplo, considere o texto "*Cultivo de arroz,banana em 1995.*". O resultado da análise léxica são as palavras "*cultivo*", "*de*", "*arroz*", "*banana*" e "*em*". Note que os caracteres de dígitos são removidos e palavras maiúsculas são convertidas em minúsculas.

Stopwords são palavras que não possuem informação relevante para a discriminação dos documentos de interesse [Baeza99]. Possíveis classes gramaticais de palavras candidatas a *stopwords* são: artigo, conjunção, contração, interjeição, preposição e pronome. A *Remoção de stopwords* tem como objetivo remover palavras que não contribuem para a categorização dos documentos. Com isso, o número de palavras a serem consideradas é reduzido. Em nossos experimentos, removemos apenas preposição do conjunto *TV* (Seção 2.1, pág. 23). Escolhemos remover apenas preposições porque, em testes preliminares, foi a opção em que os categorizadores apresentaram os melhores desempenhos de categorização.

Após a *Análise léxica* e a *Remoção de stopwords*, aplicamos o pré-processamento *Redução de dimensionalidade* (*dimensionality reduction – DR*) com o objetivo de reduzir a dimensionalidade (o número de termos) do espaço vetorial de representação dos documentos. Para isso, usamos a técnica conhecida como lematização (*lemmatization*) [Manning08], em

que as palavras dos documentos são transformadas na sua forma canônica, ou lema, isto é, o singular de um substantivo ou o infinitivo de um verbo [Antiqueira05, Cherman07]. Para implementar a lematização, utilizamos o dicionário do SCAE, que possui a forma canônica de mais de 1.200.00 de palavras do Português [SCAE08].

As palavras canônicas do conjunto *TV* que sobrevivem à Análise léxica, Remoção de *stopwords*, e Redução de dimensionalidade são denominadas *termos*. Chamamos o conjunto de termos presentes em *TV*, ou seja, o conjunto de palavras de interesse, de *Lexicon*. Com o *Lexicon*, transformamos (ou seja, indexamos) cada documento d_j de nossas bases em sua forma vetorial, $\vec{d_j} = \langle w_{1j}, w_{2j}, \ldots, w_{|T|j} \rangle$, conforme discutido na Seção 2.3. Chamamos de *Train and Test Vector* (TTV) um documento na forma vetorial.

3.4 Validação cruzada

Em problemas do mundo real, o conjunto de dados disponível para avaliar o desempenho das técnicas de categorização é limitado. Mas, para obtermos uma estimativa confiável do desempenho dos categorizadores desejamos treiná-los e testá-los com tantos documentos quanto possível. Existem muitas técnicas para tratar desse problema, mas a mais empregada na literatura, e que utilizamos neste trabalho, é a técnica n*-fold cross-validation* [Picard84].

Em n*-fold cross-validation*, o conjunto de dados é dividido em n partições mutuamente exclusivas de tamanhos aproximadamente iguais chamadas de *folds*. $n-1$ *folds* são usados para treinar, e o *fold* remanescente é usado para testar os categorizadores. Esse processo é repetido n vezes, cada vez considerando um *fold* diferente para teste. O desempenho reportado do categorizador multi-rótulo de texto segundo as métricas de avaliação de desempenho é a média dos valores das métricas obtidos em cada um dos n *folds*.

A repetição do processo de treinamento e teste permite atenuar a influência de uma amostra de treinamento e teste não representativa, tornando assim a avaliação de desempenho menos tendenciosa e mais confiável. Em experimentos da literatura, o n escolhido é freqüentemente igual a 10, pois testes extensivos sobre numerosas bases, com diferentes técnicas de categorização, têm mostrado que 10 é um número apropriado de *folds* para se obter uma estimativa confiável de desempenho [Witten05, pág. 150].

Em nossos experimentos, os 6.911 documentos da base de dados EX100 foram divididos em 10 *folds*, sendo 9 de 691 documentos e um de 692, e os 10.495 documentos da AT100 foram divididos também em 10 *folds*, sendo 9 de 1049 documentos e um de 1054. Nos experimentos com a base EX100, os categorizadores empregados foram treinados com 9 *folds* e com todos os documentos da CNAE_EX100, e testados com o décimo *fold*; enquanto que, nos experimentos com a base AT100, os categorizadores empregados foram treinados com 9 *folds* e com todos os documentos da CNAE_AT100, e testados com o décimo *fold*.

O tamanho médio do *Lexicon* para os experimentos com CNAE_EX100 e EX100 é 3609,8 termos (desvio padrão de 21,17 por conta dos diferentes *folds*), enquanto que, com CNAE_AT100 e AT100, é 5377,6 termos (desvio padrão de 19,45).

3.5 Calibração dos categorizadores

Os categorizadores apresentados no Capítulo 2 possuem parâmetros intrínsecos que devem ser ajustados (calibrados) com o objetivo de conseguir o melhor desempenho para uma determinada base de dados. Tipicamente, antes de realizar os experimentos de 10-*fold cross-validation*, os parâmetros dos categorizadores são calibrados com uma parte dos dados separada especificamente para a calibração, conhecida com dados de validação. Terminada a calibração dos categorizadores, os dados de validação são agregados aos dados de treinamento [Sebastiani02, Witten05].

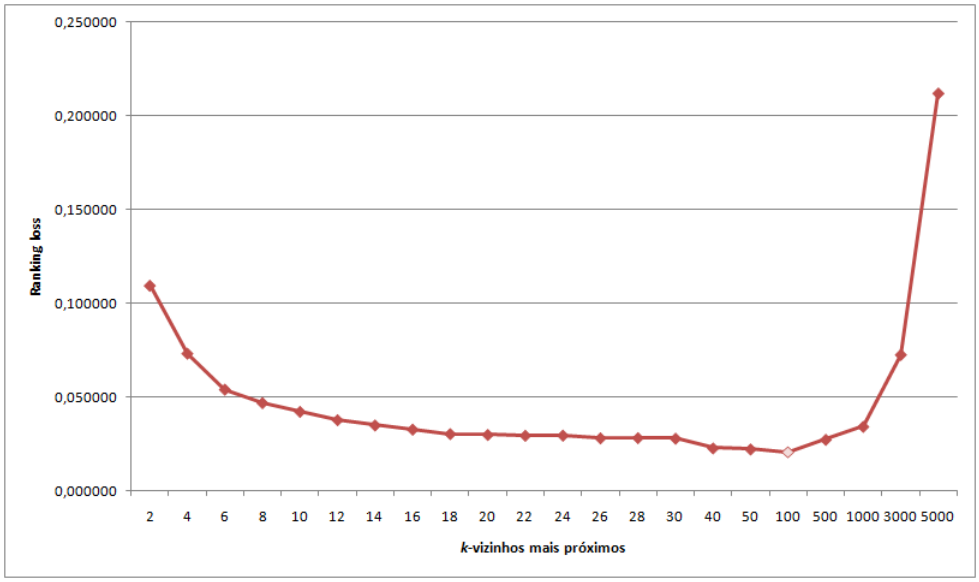
Para a calibração de cada categorizador precisamos de dados para seu treinamento e teste com o objetivo de ajuste de parâmetros. O ajuste de parâmetros é feito segundo os seguintes passos:

1. os parâmetros do categorizador são ajustados para um conjunto de valores inicial
2. o categorizador é treinado com uma parte dos dados de validação
3. o categorizador é testado com o restante dos dados de validação
4. seu desempenho medido segundo métrica específica e anotado
5. os parâmetros do categorizador são reajustados para um novo conjunto de valores
6. os passos de 2 a 5 são repetidos várias vezes e os parâmetros que produziram o melhor desempenho são escolhidos

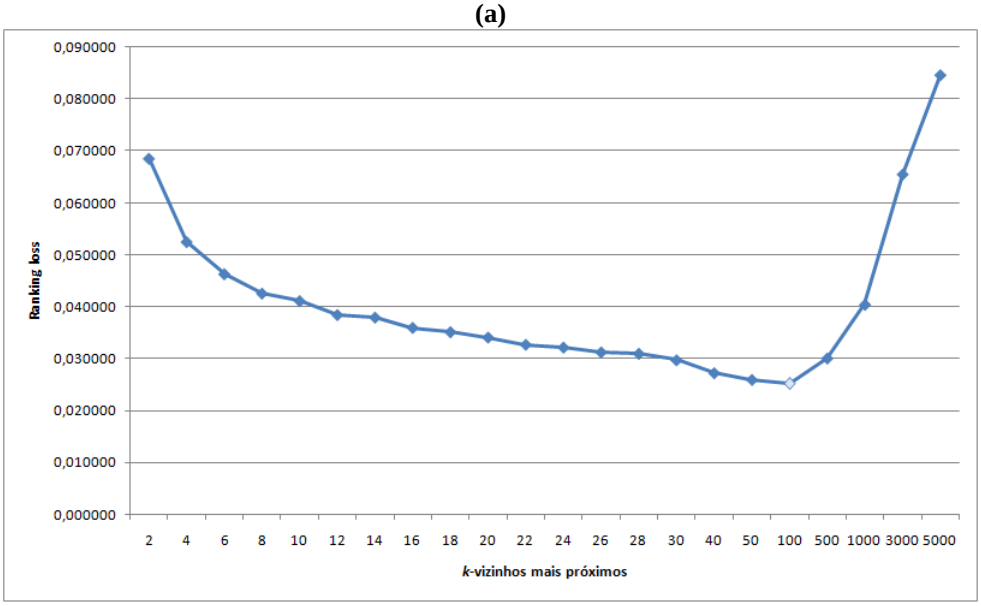
Nos nossos experimentos de calibração, escolhemos como conjunto de dados de validação os documentos de treinamento de um dos *folds* das bases de dados empregadas (EX100 ou AT100). Dividimos este conjunto de dados em 10 partes, onde as nove primeiras são utilizadas no treinamento (passo 2, acima) e a décima no teste (passo 3) dos categorizadores; testamos com apenas uma das 10 partes por conta dos custos computacionais envolvidos. A métrica empregada nos experimentos de calibração (passo 4) foi a *ranking loss*. Escolhemos esta métrica porque ela não é afetada pelo tipo de *ranking*, conforme veremos na Seção 4.1.3. Nos experimentos de calibração, todos os documentos da CNAE_EX100 e CNAE_AT100 são utilizados durante a fase de treinamento.

Os categorizadores $k\,NN$ e $ML\text{-}k\,NN$ possuem apenas um parâmetro, isto é, k (ver seções 2.4 e 2.5). No $k\,NN$, o valor de k é sempre igual ao tamanho do conjunto de dados de treinamento, $|TV|$. Assim, no categorizador $k\,NN$, o parâmetro k é fixo (não há calibração).

O categorizador $ML\text{-}k\,NN$ foi calibrado examinando seu desempenho para as ambas as bases com os seguintes valores de k: 2, 4, 6, 8, 10, 12, 14, 18, 20, 22, 24, 26, 28, 30, 40, 50, 100, 500, 1000 e 5000. A Figura 3.7 mostra os resultados obtidos no passo 4 do procedimento de calibração do $ML\text{-}k\,NN$ para as bases de dados EX100 (Figura 3.7(a)) e AT100 (Figura 3.7(b)). Nestas figuras, o eixo vertical representa o valor da métrica *ranking loss* para os diversos valores de k, e eixo horizontal os valores de k.



(a)



(b)

Figura 3.7 – Validação do *ML-* k *NN segundo a métrica ranking loss* para EX100, (a), e AT100, (b).

Conforme a Figura 3.7(a) mostra, para a base de dados EX100, este categorizador apresentou melhor desempenho segundo a métrica escolhida para $k = 100$ (ponto mais claro na Figura 3.7(a)). O mesmo ocorre com a base de dados AT100 (Figura 3.7(b)). Assim, o valor $k = 100$ foi escolhido para todos os demais experimentos com o categorizador *ML-* k *NN*.

Os categorizadores *VG-RAM WNN* e *VG-RAM WNN-COR* possuem dois parâmetros: número de neurônios ($|O|$) e número de sinapses ($|X|$). Para os dois categorizadores a calibração foi realizada com números de neurônio igual a 32, 64, 128, 256, 512 e 1024, e número de sinapses igual 256, 512, 1024 e 2048 para as bases de dados EX100 e AT100.

A Figura 3.8 e a Figura 3.9 apresentam os resultados do processo de validação do *VG-RAM WNN* para a as bases EX100 e AT100, respectivamente. Nestas figuras, o eixo vertical representa o valor da métrica *ranking loss* para os diversos valores de $|O|$ do eixo horizontal e cada curva está associada a um valor de $|X|$, conforme indicado na legenda de cada figura.

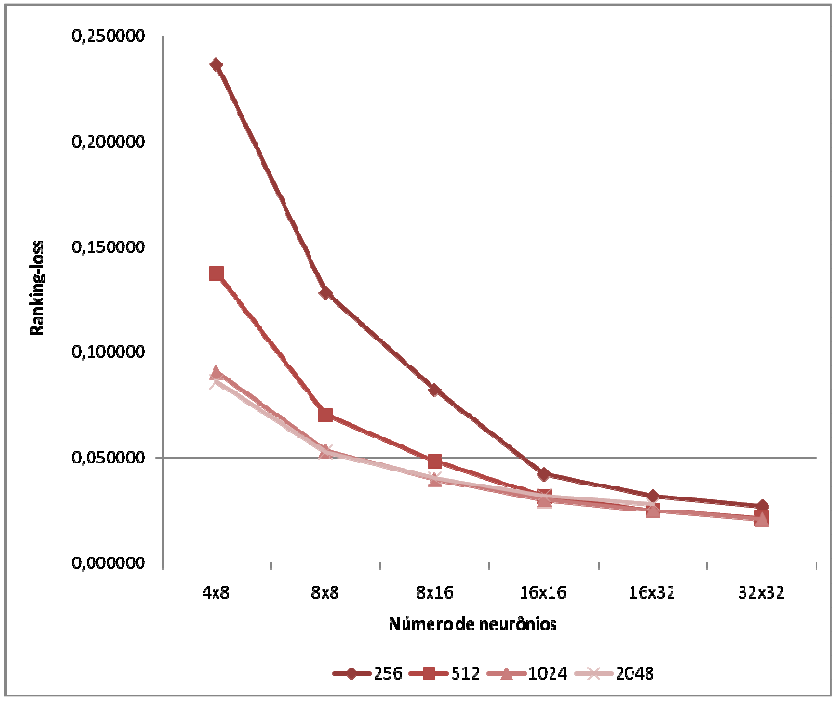


Figura 3.8 – Validação do *VG-RAM WNN* na base EX100.

Conforme a Figura 3.8 mostra, este categorizador apresentou melhor desempenho segundo a métrica escolhida para $|O|$ igual a 1024 $(32x32)$ neurônios, mas não está claro na figura qual o melhor número de sinapses. A Tabela 3.1 mostra o valor de *ranking loss* (última coluna) para os diferentes valores de $|X|$ examinados quando $|O|$ é igual a $32x32$ (último ponto à esquerda de cada curva da Figura 3.8). Como mostra a Tabela 3.1, o menor valor de *ranking loss* ocorre com $|X| = 1024$ sinapses. Note que, para $32x32$ neurônios e 2048 sinapses não existe ponto no gráfico da Figura 3.8 ou valor na Tabela 3.1, pois a quantidade de memória necessária para armazenar os dados de treinamento com esta configuração excede a capacidade de endereçamento de um processador de 32 bits, 4 *giga bytes*. Assim, para a base de dados EX100, os valores $|O| = 32x32$ e $|X| = 1024$ foram escolhidos para todos os demais experimentos com o categorizador *VG-RAM WNN*.

Tabela 3.1 – Validação para *VG-RAM WNN* na EX100 para 32x32 neurônios.

Sinapses	*Ranking* loss
256	0,027144
512	0,021134
1024	0,020765
2048	-

De acordo com a Figura 3.9, o *VG-RAM WNN* apresentou melhor desempenho segundo a métrica *ranking loss* para a base AT100 com 1024 ($32x32$) neurônios, mas, novamente, não está claro na figura qual o melhor número de sinapses. Como mostra a Tabela 3.2, o menor valor de *ranking loss* ocorre com 512 sinapses. Assim, para a base de dados AT100, os valores $|O| = 32x32$ e $|X| = 512$ foram escolhidos para todos os demais experimentos com o categorizador *VG-RAM WNN*.

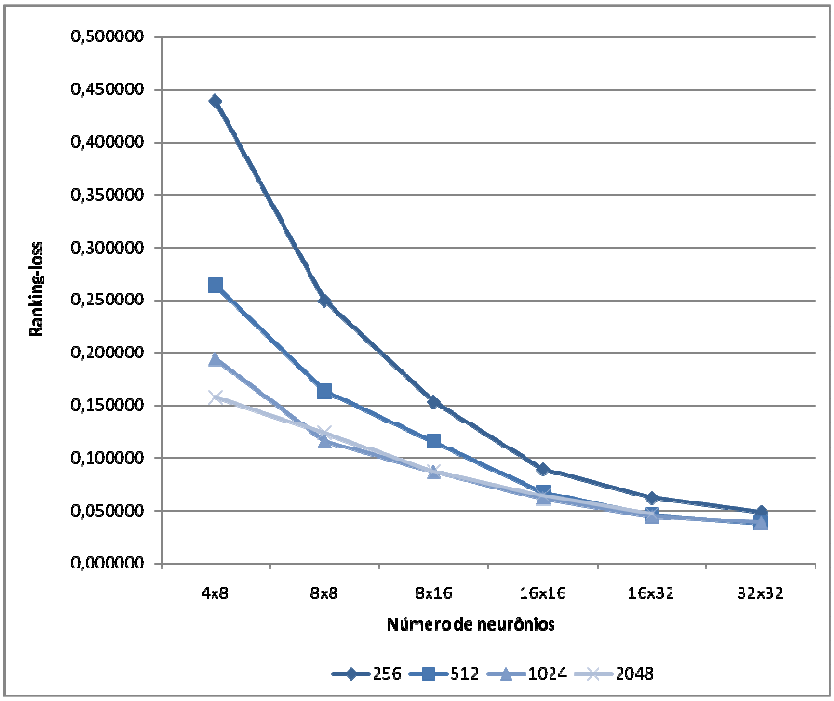


Figura 3.9 – Validação do *VG-RAM WNN* na base AT100.

Tabela 3.2 – Validação para *VG-RAM WNN* na AT100 para 32x32 neurônios.

Sinapses	*Ranking* loss
256	0,048425
512	0,037989
1024	0,038632
2048	-

A Figura 3.10 e a Figura 3.11 apresentam os resultados do processo de validação do *VG-RAM WNN-COR* para as bases EX100 e AT100, respectivamente.

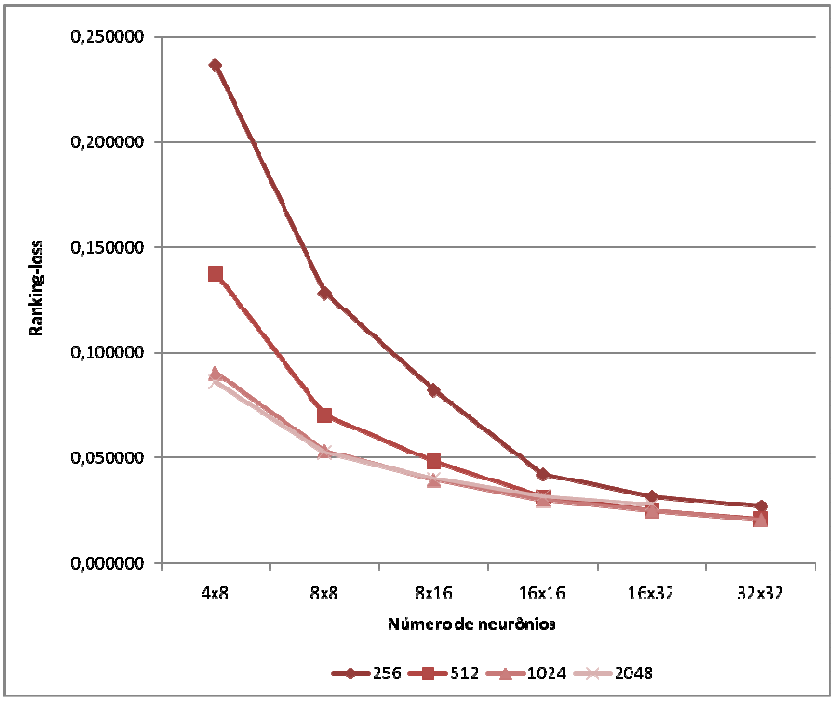


Figura 3.10 – Validação do *VG-RAM WNN-COR* na base EX100.

Conforme mostra a Figura 3.10, este categorizador apresentou melhor desempenho segundo a métrica escolhida para 1024 $(32x32)$ neurônios, mas, mais uma vez, não está claro na figura qual o melhor número de sinapses. Como mostra a Tabela 3.3, o melhor número de sinapses é 512. Assim, para a base de dados EX100, os valores $|O|=32x32$ e $|X|=512$ foram escolhidos para todos os demais experimentos com o categorizador *VG-RAM WNN-COR*.

Tabela 3.3 – Validação para *VG-RAM WNN-COR* na EX100 para 32x32 neurônios.

Sinapses	***Ranking* loss**
256	0,024758
512	0,020754
1024	0,021162
2048	0,022277

De acordo com a Figura 3.11, o *VG-RAM WNN-COR* apresentou melhor desempenho segundo a métrica *ranking loss* para 1024 $(32x32)$ neurônios e 1024 sinapses. Assim, para a base de dados AT100, os valores $|O|=32x32$ e $|X|=1024$ foram escolhidos para todos os demais experimentos com o categorizador *VG-RAM WNN-COR*.

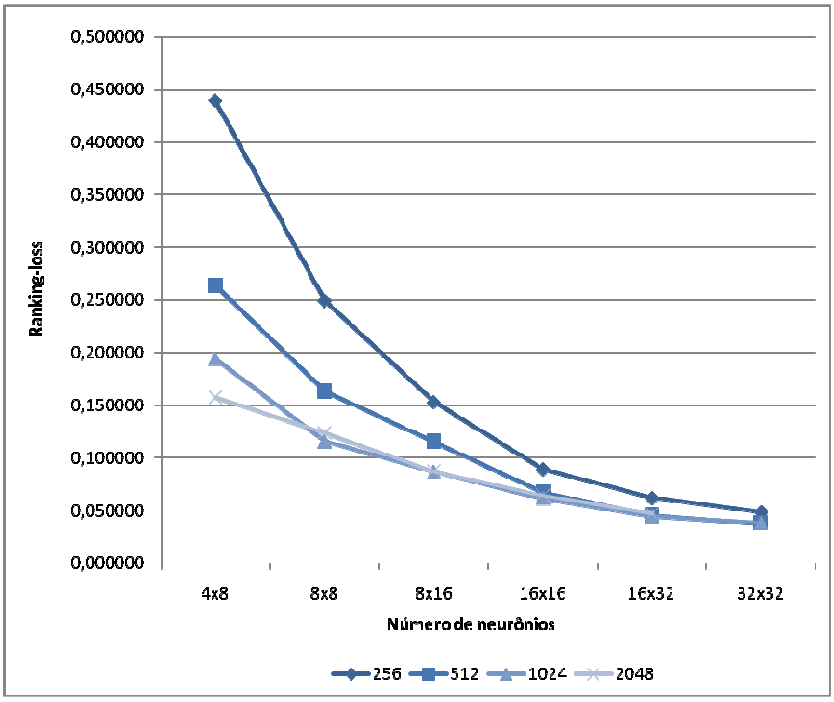


Figura 3.11 – Validação do *VG-RAM WNN-COR* na base AT100.

A Tabela 3.4 sumariza os parâmetros escolhidos para cada categorizador (primeira coluna à esquerda) para as bases de dados EX100 (coluna do meio) e AT100 (última coluna à direita).

Tabela 3.4 – Sumário das escolhas dos parâmetros dos categorizadores na validação para EX100 e AT100.

Categorizador	**Bases de dados**	
	EX100	**AT100**
ML-k NN	$k = 100$	$k = 100$
VG-RAM WNN	$\| O \| = 32x32$ $\| X \| = 1024$	$\| O \| = 32x32$ $\| X \| = 512$
VG-RAM WNN-COR	$\| O \| = 32x32$ $\| X \| = 512$	$\| O \| = 32x32$ $\| X \| = 1024$

3.6 Verificação estatística do impacto do *ranking* sobre as métricas de categorização multi-rótulo de texto

Para verificar se diferentes tipos de *ranking* afetam o desempenho de categorizadores multi-rótulo segundo diferentes métricas, usamos um teste estatístico de hipótese – o teste t de *Student* [Student08] – que apresentamos a seguir.

Considere duas amostras, $X = \{x_1, x_2, ..., x_{10}\}$ e $Y = \{y_1, y_2, ..., y_{10}\}$, de medidas de desempenho, segundo uma determinada métrica, obtidas empregando-se os *rankings* hipotéticos $A1$ e $A2$, respectivamente, onde x_i corresponde ao desempenho do categorizador para o *fold* i com o *ranking* $A1$ e y_i com o *ranking* $A2$. O que desejamos determinar é se o desempenho medido com a métrica empregando o *ranking* $A1$ é diferente do desempenho medido com a métrica empregando o *ranking* $A2$. Para verificar isso, definimos como hipótese nula, representada por H_0, que o desempenho medido com a métrica empregando o *ranking* $A1$ é **igual** ao medido com a métrica empregando o *ranking* $A2$; e como hipótese alternativa, representada por H_1, que o desempenho medido com a métrica empregando o *ranking* $A1$ é **diferente** do desempenho medido com a métrica empregando o *ranking* $A2$.

Na literatura [Mitchell97, Witten05, Zhang06], o método estatístico utilizado para este tipo de verificação é o teste t de *Student* (*Student's* t*-test*). O teste t de *Student* avalia a significância estatística da diferença entre as médias de duas amostras independentes [Hair05]. A estatística t examinada neste teste é computada dividindo-se a média das diferenças, $\bar{d}$, das duas amostras pelo seu erro padrão, $\sigma_d / \sqrt{n}$, conforme Equação (3.2):

$$t = \frac{\bar{d}}{\sigma_d / \sqrt{n}} \tag{3.2}$$

onde σ_d é o desvio padrão das diferenças e n é o tamanho das duas amostras, que, no nosso caso, é o número de *folds*, isto é, $n = 10$.

Devido ao tamanho das amostras ($n < 30$), a estatística t possui uma distribuição t de *Student* com $n-1$ graus de liberdade. A Figura 3.12 mostra graficamente um exemplo de distribuição de *Student*. O eixo horizontal do gráfico corresponde à estatística t e o eixo vertical à função de distribuição de probabilidade ($F(t) = P[T \leq t]$). As áreas sombreadas do gráfico são as regiões de rejeição da hipótese nula H_0 para um valor t_{crit}, isto é, aceitação da hipótese alternativa H_1. A área entre as áreas sombreadas é a região de aceitação da hipótese nula para um valor t_{crit}.

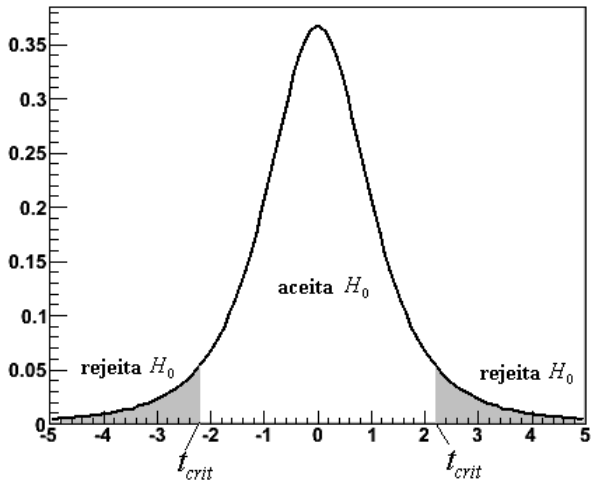


Figura 3.12 - Exemplo gráfico da distribuição t de Student.

Para verificar se as médias das amostras são estatisticamente diferentes, compara-se a estatística t obtida pela Equação (3.2) com o valor crítico da estatística t (t_{crit}). Se o valor da estatística t é maior que t_{crit} ou menor que $-t_{crit}$, a hipótese nula é rejeitada (região sombreada do gráfico da Figura 3.12). Caso contrário, a hipótese nula é aceita (Figura 3.12). O valor t_{crit} é determinado de acordo com nível de significância α desejado e o número de graus de liberdade da distribuição de *Student* (Tabela 3.5), que no nosso caso é $n = 9$. O nível de significância α indica o nível de probabilidade de aceitação da hipótese alternativa quando na verdade é para ser rejeitada, isto é, uma probabilidade de erro de α.

A Tabela 3.5 mostra os valores t_{crit} (coluna à direita) com os respectivos níveis de significância α (coluna à esquerda) para distribuição de *Student* com 9 graus de liberdade.

Tabela 3.5 - Níveis de significância α com os respectivos valores t_{crit} para distribuição de *Student* com 9 graus de liberdade.

α	t_{crit}
0,1%	*4,2969*
0,5%	*3,2498*
1%	*2,8214*
2,5%	*2,2622*
5%	*1,8331*
10%	*1,3830*

O procedimento que empregamos para verificar se o desempenho medido com a métrica empregando o *ranking* $A1$ é diferente do desempenho medido com a métrica empregando o *ranking* $A2$, isto é, se a hipótese nula é rejeitada ou não, é como abaixo:

1. escolhemos a métrica de avaliação de desempenho
2. definimos o *ranking* $A1$ como o *ranking* Ordinal Aleatório
3. definimos o *ranking* $A2$ como o *ranking* Denso para comparação com o *ranking* Ordinal Aleatório
4. escolhemos um nível de significância α
5. obtemos o valor t_{crit} de acordo com Tabela 3.5
6. a estatística t é computada conforme Equação (3.2) e anotada
7. o valor da estatística t é comparado ao valor t_{crit}
8. os passos 2 a 7 são repetidos substituindo o *ranking* Denso pelo o *ranking* Padrão
9. os passos 2 a 7 são repetidos substituindo o *ranking* Denso pelo o *ranking* Modificado
10. os passos 1 a 9 são repetidos para outra métrica de avaliação

Em nossos experimentos, avaliamos a significância estatística por meio do teste t pareado bicaudal [Hair05] entre o desempenho de um categorizador medido com a métrica empregando o *ranking* Ordinal Aleatório (passo 2, acima) e o desempenho medido com a métrica empregando os tipos de *ranking* Denso, Padrão, e Modificado (passo 3). Escolhemos um nível de significância $\alpha = 5\%$ (passo 4), mas, como utilizamos o teste t bicaudal, escolhemos o valor t_{crit} correspondente a $\alpha = 2{,}5\%$ na Tabela 3.5, isto é, $t_{crit} = \pm 2{,}2622$ (passo 5) [Hair05]. Calculamos a estatística t (passo 6) e comparamos com o valor t_{crit} (passo 7). Se a estatística t é maior que o valor t_{crit} ou menor que $-t_{crit}$, a hipótese nula é rejeitada; caso contrário, a hipótese nula é aceita.

Utilizamos em nossos experimentos o teste t pareado (*paired t-test*) bicaudal, pois (i) os experimentos de 10-*fold cross-validation* são realizados sobre os mesmos conjuntos de documentos de treinamento e de teste para cada categorizador [Mitchell97, Witten95], e (ii) porque não sabemos previamente se o desempenho medido com a métrica empregando o *ranking* Ordinal Aleatório é maior ou menor que com o desempenho medido com a métrica empregando os demais tipos de *ranking* em estudo [Witten05]. Verificamos também, por amostragem, que os erros $(d_i = x_i - y_i)$ entre as amostras de desempenho dos categorizadores, obtidas segundo a técnica de 10-*fold cross-validation* para cada métrica de avaliação, seguem uma distribuição normal, o que permite aplicar o *teste t de Student* em

nossos experimentos. Vale destacar que, segundo Hull [Hull93], frequentemente é útil o *teste t* mesmo quando os erros supramencionados não seguem uma distribuição normal.

4 AVALIAÇÃO EXPERIMENTAL DO EFEITO DO *RANKING* NAS MÉTRICAS DE AVALIAÇÃO DE DESEMPENHO

Neste capítulo, examinamos o impacto dos tipos de *ranking* Ordinal Aleatório, Denso, Padrão e Modificado no desempenho dos categorizadores k *NN*, *ML-*k *NN*, *VG-RAM WNN* e *VG-RAM WNN-COR* no contexto de categorização de descrições de atividades econômicas de empresas brasileiras segundo a CNAE [CNAE03]. O desempenho dos categorizadores foi medido segundo as seguintes métricas de avaliação de desempenho de categorizadores multi-rótulo: *one-error* [Schapire99], *coverage* [Schapire00], *ranking loss* [Schapire99], *average precision* [Schapire00, Manning08], *R-precision* [Manning08], *Hamming loss* [Schapire99], *exact match* [Kazawa05], *precision* [Sebastiani02], [Manning08], *recall* [Sebastiani02, Manning08] e F_1 [Sebastiani02, Manning08].

A definição de algumas métricas foi reformulada para comportar o tratamento de empates nos *rankings* Denso, Padrão e Modificado. Segundo Manning [Manning08], as métricas de avaliação de desempenho podem ser classificadas em dois grupos:

i. **métricas de avaliação para conjuntos ordenados**, que avaliam todo o *ranking* de categorias derivado da função $f(.,.)$, dentre as quais incluem *one-error* [Schapire99], *coverage* [Schapire00], *ranking loss* [Schapire99], *average precision* [Schapire00, Manning08], *R-precision* [Manning08];

ii. **métricas de avaliação para conjuntos não ordenados**, que avaliam o conjunto exato de categorias predito, $\hat{C}_j$, para o documento de teste d_j, dentre as quais incluem *Hamming loss* [Schapire99], *exact match* [Kazawa05], *precision* [Sebastiani02], [Manning08], *recall* Sebastiani02, Manning08] e F_β [Sebastiani02, Manning08].

Nas seções a seguir, apresentamos, primeiramente, a formulação original de cada métrica e, em seguida, a nossa proposta de reformulação juntamente com as justificativas. Apresentamos também, os resultados experimentais mostrando o efeito dos tipos de *ranking*

considerados no desempenho dos categorizadores segundo cada métrica para as bases de dados EX100 e AT100.

4.1 Métricas de avaliação para conjuntos ordenados

4.1.1 One-error

A métrica ***one-error*** ($one-error_j$) avalia se a categoria no topo do *ranking* está presente no conjunto das categorias pertinentes C_j do documento de teste d_j. A formulação original é apresentada na Equação (4.1) [Schapire99].

$$one-error_j = \begin{cases} 0 \quad se \quad [\arg\max_{c_i \in C} f(d_j, c_i)] \in C_j \\ 1 \quad caso \quad contrário \end{cases} \tag{4.1}$$

onde $[\arg\max_{c_i \in C} f(d_j, c_i)]$ retorna **a categoria no topo do *ranking*** para o documento de teste d_j.

A métrica *one-error* avalia se, e somente se, uma categoria pertinente está no topo do *ranking*, não considerando a questão de empates entre categorias no topo. A nossa proposta de reformulação da métrica *one-error*, que chamamos de *one-error**, por outro lado, avalia se **todas as categorias no topo do *ranking*** estão presentes no conjunto de categorias pertinentes do documento de teste d_j. A reformulação da métrica é apresentada na Equação (4.2).

$$one-error_j* = \begin{cases} 0 \quad se \quad \{[\arg\max_{c_i \in C} f(d_j, c_i)]\} \subset C_j \\ 1 \quad caso \quad contrário \end{cases} \tag{4.2}$$

onde $\{[\arg\max_{c_i \in C} f(d_j, c_i)]\}$ retorna **uma ou mais categorias empatadas no topo** do *ranking* para o documento de teste d_j.

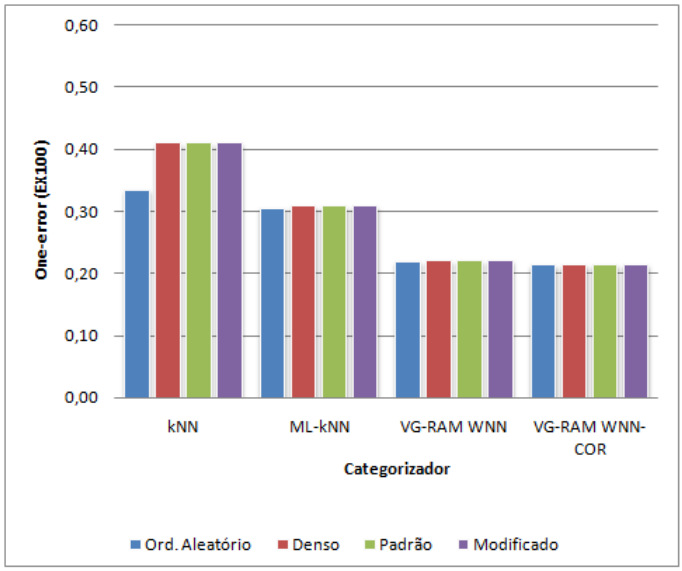
Na redefinição da métrica, se existe mais de uma categoria no topo do *ranking* e todas são pertinentes, o valor de $one-error_j*$ é zero, caso contrário, é um. É importante observar que a métrica *one-error** é equivalente à definição original se não houver empates, isto é, ela generaliza *one-error* para os casos de empates, penalizando o categorizador que não é capaz de atribuir valores distintos de $f(d_j,.)$ para categorias de interesse no topo do *ranking*.

O desempenho global, isto é, o sumário do valor da métrica para o conjunto Te, é obtido pela Equação (4.3). Quanto menor o valor de *one-error**, melhor o desempenho do categorizador. O desempenho é perfeito ocorre quando $one-error^* = 0$.

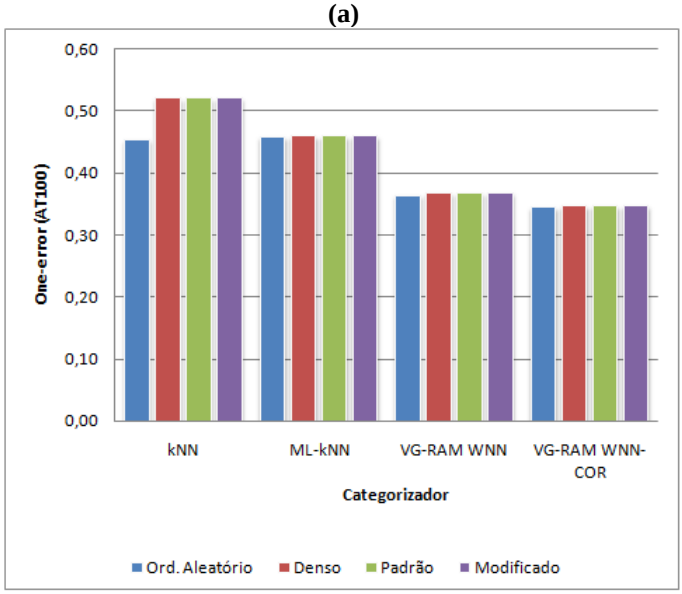
$$one-error^* = \frac{1}{|Te|}\sum_{j=1}^{|Te|} one-error_j \quad (4.3)$$

Pode-se utilizar o *ranking* Ordinal Aleatório para ocultar os empates, mas, no caso da métrica *one-error*, isso pode favorecer categorizadores por mero acaso (uma categoria correta pode ser ranqueada no topo por acaso mesmo que ela esteja empatada com várias outras categorias). Para mostrar isso, realizamos os experimentos apresentados na Figura 4.1.

A Figura 4.1 mostra de forma gráfica o impacto de cada tipo de *ranking* na métrica *one-error** para a base EX100 (Figura 4.1(a)) e AT100 (Figura 4.1(b)). Em cada um dos gráficos da Figura 4.1, existem quatro conjuntos de barras, um para cada categorizador empregado, onde a amplitude de cada barra indica o valor de *one-error** (média dos 10 *folds*) para cada um dos diversos tipos de *ranking* em estudo. Em cada conjunto de barras, da esquerda para a direita, a primeira barra indica o *ranking* Ordinal Aleatório (Ord. Aleatório), a segunda o *ranking* Denso, a terceira o *ranking* Padrão, e a quarta o *ranking* Modificado (ver legenda nos gráficos da figura).



(a)



(b)

Figura 4.1 – Resultado da métrica *one-error para a base EX100, (a), e AT100, (b). Quanto menor, melhor.**

Como as barras do gráfico da Figura 4.1(a) mostram, o desempenho segundo *one-error** do categorizador *k NN* com a base EX100 é impactado pelo tipo de *ranking* empregado. O valor de *one-error** com o *ranking* Ordinal Aleatório é significativamente menor que com o *ranking* Denso, Padrão e Modificado (teste t pareado bicaudal com nível de significância 5%). O mesmo ocorre com a base de dados AT100 (Figura 4.1(b)). O valor de *one-error** com os tipos de *ranking* Denso, Padrão e Modificado são os mesmos para ambas as bases, visto que a métrica *one-error** avalia o topo do *ranking*, e o topo dos *rankings* Denso, Padrão e Modificado são iguais, ainda que existam empates.

Os resultados obtidos com o categorizador $k\,NN$ sugerem que o *ranking* mais apropriado é o Denso, o Padrão ou o Modificado, já que estes penalizam o categorizador no caso de empates. O *ranking* Ordinal Aleatório, possivelmente o mais freqüentemente utilizado na literatura, não é o mais apropriado, pois favorece categorizadores que geram *rankings* com empates no topo.

A análise do desempenho do categorizador $ML\text{-}k\,NN$ segundo a métrica *one-error** mostra que o desempenho deste categorizador também é significativamente afetado pelo tipo de *ranking* para a base de dados EX100, muito embora o conjunto de barras da Figura 4.1(a) associado a este categorizador não permita ver isso claramente. O valor de *one-error** deste categorizador com o *ranking* Ordinal Aleatório é significativamente menor que com os *rankings* Denso, Padrão e Modificado, conforme mostra a Tabela 4.1, detalhada a seguir.

A Tabela 4.1 apresenta a estatística t associada à comparação do desempenho dos categorizadores com o *ranking* Ordinal Aleatório, com o desempenho dos categorizadores com os demais tipos de *ranking* em estudo segundo a métrica *one-error** para cada uma das bases de dados. Se a estatística t é maior que o valor $t_{crit} = 2{,}2622$ ou menor que $t_{crit} = -2{,}2622$, o valor de *one-error** com o *ranking* Ordinal Aleatório é significativamente maior ou significativamente menor que com o *ranking* Denso, Padrão ou Modificado; caso contrário, o valor de *one-error** com o *ranking* Ordinal Aleatório não é significativamente diferente do valor de *one-error** com o *ranking* Denso, Padrão ou Modificado (teste t pareado bicaudal com nível de significância 5%). Na tabela, a estatística t com a cor vermelha indica uma diferença significativa a menor, e a cinza que não há diferença significativa entre o desempenho do categorizador com o *ranking* Ordinal Aleatório e o outro tipo de *ranking* correspondente à posição na tabela.

Tabela 4.1 – A estatística *t* da comparação do desempenho com o *ranking* Ordinal Aleatório, com o desempenho com os demais tipos de *ranking* segundo *one-error para as bases EX100 e AT100.**

Base de dados	Categorizador	Denso	Padrão	Modificado
EX100	kNN	*-19,2686*	*-19,2686*	*-19,2686*
	ML-kNN	*-6,7342*	*-6,7342*	*-6,7342*
	VG-RAM WNN	*-3,2501*	*-3,2501*	*-3,2501*
	VG-RAM WNN-COR	*-2,2359*	*-2,2359*	*-2,2359*
AT100	kNN	*-25,5917*	*-25,5917*	*-25,5917*
	ML-kNN	*-2,2118*	*-2,2118*	*-2,2118*
	VG-RAM WNN	*-6,9214*	*-6,9214*	*-6,9214*
	VG-RAM WNN-COR	*-3,3611*	*-3,3611*	*-3,3611*

Como a Tabela 4.1 mostra, o desempenho do categorizador *ML-k NN* não é afetado pelo tipo de *ranking* para a base de dados AT100. Ou seja, o valor de *one-error** deste categorizador com o *ranking* Ordinal Aleatório não é significativamente diferente que com os *rankings* Denso, Padrão e Modificado. As três últimas barras do gráfico da Figura 4.1(a) associadas a este categorizador mostram que o valor de *one-error** com os tipos de *ranking* Denso, Padrão e Modificado são os mesmos no caso da base EX100, o que também ocorre com a base AT100 – Figura 4.1(b).

Como a Figura 4.1 mostra, o impacto do tipo de *ranking* no desempenho dos categorizadores *VG-RAM WNN* e *VG-RAM WNN-COR* é similar àquele observado no categorizador *ML-k NN*, muito embora não seja idêntico. De acordo com a Tabela 4.1, o desempenho do categorizador *VG-RAM WNN* segundo a métrica *one-error** é significativamente afetado pelo tipo de *ranking* para a base de dados EX100. O valor de *one-error** deste categorizador com o *ranking* Ordinal Aleatório é significativamente menor que com os *rankings* Denso, Padrão e Modificado. O mesmo ocorre para a base de dados AT100. A estatística t mostrada na Tabela 4.1 com os tipos de *ranking* Denso, Padrão e Modificado são iguais para a base de dados EX100 neste categorizador, mostrando que o valor de *one-error** é equivalente com os tipos de *ranking* empregados. Isso também ocorre com a base AT100. Por outro lado, a análise de desempenho do categorizador *VG-RAM WNN-COR* segundo a métrica *one-error** mostra que o desempenho deste categorizador não é afetado pelo tipo de *ranking* para a base de dados EX100. No entanto, para a base de dados AT100, o desempenho deste categorizador segundo a métrica *one-error** é significativamente afetado pelo tipo de *ranking*. O valor de *one-error** deste categorizador com o *ranking* Ordinal Aleatório é significativamente menor que com os *rankings* Denso, Padrão e Modificado. A estatística t da Tabela 4.1 com os tipos de *ranking* Denso, Padrão e Modificado são iguais para a base de dados EX100, e o mesmo ocorre com a base AT100.

Os resultados obtidos com os categorizadores *k NN*, *ML-k NN*, *VG-RAM WNN* e *VG-RAM WNN-COR* para a métrica *one-error** mostram que o *ranking* mais apropriado é o Denso, o Padrão, ou o Modificado. O *ranking* Ordinal Aleatório não é o mais apropriado, pois favoreceu os categorizadores que geraram *rankings* com empates no topo entre as categorias de interesse, com exceção do categorizador *VG-RAM WNN-COR* para a base EX100 e do categorizador *ML-k NN* para a base AT100.

4.1.2 Coverage

A métrica ***coverage*** (*coverage*$_j$) mede quantas posições no *ranking* de categorias do documento de teste d_j precisamos descer, de modo a abranger todas as categorias pertinentes. A formulação original de Schapire et al. [Schapire00] é apresentada na Equação (4.4).

$$coverage_j = \max_{c_i \in C_j} r(d_j, c_i) - 1 \quad (4.4)$$

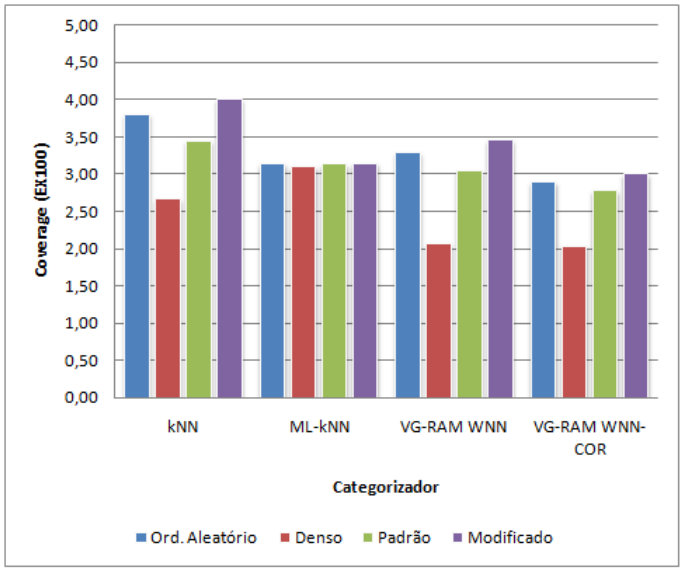
onde $\max_{c_i \in C_j} r(d_j, c_i)$ retorna a posição mais baixa do *ranking* (de índice mais alto) que contém uma categoria pertinente a d_j. O desempenho global é dado pela Equação (4.5).

$$coverage = \frac{1}{|Te|} \sum_{j=1}^{|Te|} coverage_j \quad (4.5)$$

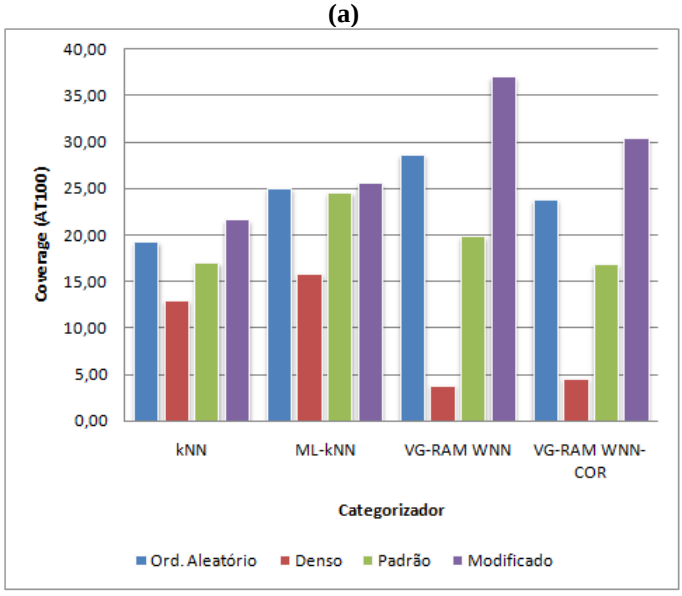
Quanto menor o valor de *coverage*, melhor o desempenho do categorizador. O desempenho ideal ocorre quando $coverage = \frac{1}{|Te|} \sum_{j=1}^{|Te|} (|C_j| - 1)$.

A definição da métrica *coverage* não precisa ser reformulada, pois a mesma avalia o categorizador de acordo com a posição do *ranking*, o que permite empregar os *rankings* Denso, Padrão e Modificado.

A Figura 4.2 mostra de forma gráfica o impacto de cada tipo de *ranking* na métrica *coverage* para a base EX100 (Figura 4.2(a)) e AT100 (Figura 4.2(b)). Esta figura segue o mesmo formato da Figura 4.1.



(a)



(b)

Figura 4.2 – Resultado da métrica *coverage* para a base EX100, (a), e AT100, (b). Quanto menor, melhor.

Conforme as barras do gráfico da Figura 4.2(a) mostram, o valor da métrica *coverage* do categorizador *k NN* com a base EX100 é visivelmente diferente para cada tipo de *ranking* empregado. O valor de *coverage* com o *ranking* Ordinal Aleatório é significativamente maior que com o *ranking* Denso e Padrão, e menor que com o *ranking* Modificado (teste *t* pareado bicaudal com nível de significância 5%). O mesmo ocorre com os categorizadores *VG-RAM WNN* e *VG-RAM WNN-COR* com esta base, e com todos os categorizadores com a base de dados AT100 (Figura 4.2(b)).

A análise do desempenho do categorizador *ML-k NN* segundo a métrica *coverage* para a base de dados EX100 (Figura 4.2(a)) mostra que o desempenho deste categorizador também é significativamente afetado pelo tipo de *ranking* empregado. O valor de *coverage* com o

ranking Ordinal Aleatório é significativamente maior que com o *ranking* Denso e Padrão, e menor que com o *ranking* Modificado, conforme discutido abaixo.

Como na Tabela 4.1, a Tabela 4.2 mostra a estatística t associada à comparação do desempenho dos categorizadores com o *ranking* Ordinal Aleatório, com o desempenho com os demais tipos de *ranking* em estudo segundo a métrica *coverage* para cada uma das bases de dados. Uma diferença entre a Tabela 4.1 e a Tabela 4.2 é a cor verde, que indica uma diferença significativa a maior (teste t pareado bicaudal com um nível de significância 5%) entre o desempenho do categorizador com o *ranking* Ordinal Aleatório e o outro tipo de *ranking* correspondente à posição na tabela.

Tabela 4.2 – A estatística *t* da comparação do desempenho com o *ranking* Ordinal Aleatório, com o desempenho com os demais tipos de *ranking* segundo *coverage* para as bases EX100 e AT100.

Base de dados	Categorizador	Denso	Padrão	Modificado
EX100	kNN	*31,9235*	*9,0984*	*-5,5330*
	ML-kNN	*9,2827*	*6,3923*	*-6,5143*
	VG-RAM WNN	*17,4571*	*10,5727*	*-13,0325*
	VG-RAM WNN-COR	*14,8499*	*11,1590*	*-15,7044*
AT100	kNN	*18,5644*	*8,3874*	*-11,5611*
	ML-kNN	*14,8490*	*9,9785*	*-14,7447*
	VG-RAM WNN	*15,1988*	*19,5440*	*-20,1028*
	VG-RAM WNN-COR	*21,6349*	*15,9926*	*-29,3762*

Conforme mostra a Tabela 4.2, o desempenho de todos os categorizadores segundo a métrica *coverage* é impactado pelos tipos de *ranking* Denso, Padrão e Modificado para as bases de dados EX100 e AT100. O valor de *coverage* destes categorizadores com o *ranking* Ordinal Aleatório é significativamente maior que com os *rankings* Denso e Padrão e menor que com o *ranking* Modificado.

Os resultados obtidos com os categorizadores mostram que o *ranking* mais apropriado segundo a métrica *coverage* é o Modificado, pois os *rankings* Ordinal Aleatório, Denso e Padrão favoreceram os categorizadores que produziram empates entre as categorias de interesse presentes nos *rankings*.

4.1.3 Ranking loss

A métrica ***ranking loss*** ($ranking - loss_j$) avalia a fração de pares de categorias $\langle c_i, c_k \rangle$, onde $c_i \in C_j$ e $c_k \in \overline{C}_j$, que estão reversamente ordenados ($f(d_j, c_i) \leq f(d_j, c_k)$)

no *ranking* de categorias do documento de teste d_j. A formulação original de Schapire et al. [Schapire00] é apresentada na Equação (4.6), abaixo:

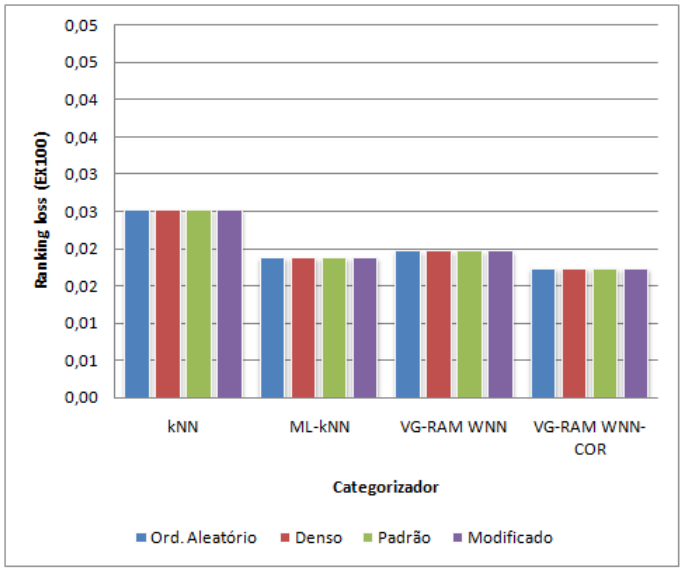
$$ranking-loss_j = \frac{1}{|C_j||\overline{C}_j|}\left|\left\{(c_i,c_k)\mid f(d_j,c_i) \le f(d_j,c_k),\ \ (c_i,c_k)\in C_j\times\overline{C}_j\right\}\right| \qquad (4.6)$$

onde $\overline{C}_j$ é o conjunto complementar de C_j em C.

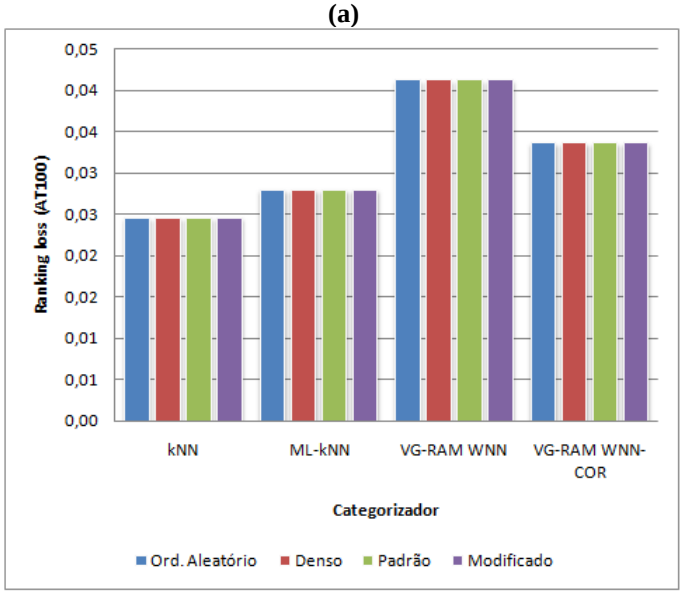
O desempenho global é computado pela Equação (4.7). Quanto menor o valor de *ranking loss*, melhor o desempenho do sistema de categorização. O desempenho é ideal quando $ranking-loss = 0$.

$$ranking-loss = \frac{1}{|Te|}\sum_{j=1}^{|Te|} ranking-loss_j \qquad (4.7)$$

A Figura 4.3 mostra de forma gráfica o impacto de cada tipo de *ranking* na métrica *ranking loss* para a base EX100 (Figura 4.3(a)) e AT100 (Figura 4.3(b)). Esta figura segue o mesmo formato da Figura 4.1.



(a)



(b)

Figura 4.3 – Resultado da métrica *ranking loss* para a base EX100, (a), e AT100, (b). Quanto menor, melhor.

Conforme as barras do gráfico da Figura 4.3(a) mostram, o valor da métrica *ranking loss* do categorizador $k\,NN$ com a base EX100 é indiferente ao tipo de *ranking* empregado. Ou seja, o valor da métrica *ranking loss* com o *ranking* Ordinal Aleatório não é estatisticamente diferente (teste t pareado bicaudal com nível de significância 5%) que com os *rankings* Denso, Padrão e Modificado. O mesmo acontece com a base de dados AT100 (Figura 4.3(b)). Esta indiferença no valor da métrica *ranking loss* quanto a tipo de *ranking* empregado já era esperada, pois esta métrica mede a fração de pares de categorias reversamente ordenados com base na função $f(.,.)$, em vez das posições de *rankings* geradas pelos tipos de *ranking* em estudo.

O comportamento observado no categorizador $k\,NN$ quanto ao tipo de *ranking* também acontece com os categorizadores *ML-*$k\,NN$, *VG-RAM WNN* e *VG-RAM WNN-COR* para a base de dados EX100(Figura 4.3(a)) e para a AT100(Figura 4.3(b)).

Os resultados obtidos com os categorizadores $k\,NN$, *ML-*$k\,NN$, *VG-RAM WNN* e *VG-RAM WNN-COR* para a métrica *ranking loss* mostram que o desempenho dos categorizadores é indiferente ao tipo de *ranking* empregado. Então, os tipos de *ranking* Ordinal Aleatório, Denso, Padrão ou Modificado são apropriados para a métrica *ranking loss*.

4.1.4 Average precision

A métrica ***average precision*** ($avg-precision_j$) avalia a média das precisões computadas ao truncar o *ranking* de categorias em cada categoria $c_i \in C_j$. A formulação original de Schapire et al. [Schapire00] é apresentada na Equação (4.8).

$$avg-precision_j = \frac{1}{|C_j|}\sum_{k=1}^{|C_j|}\frac{|\hat{C}_j^k \cap C_j|}{|\hat{C}_j^k|} \tag{4.8}$$

onde $\hat{C}_j^k$ é o conjunto de categorias preditas que vão do topo do *ranking* até a posição k do *ranking*. Se $f(d_j, c_i) = 0$ para a categoria $c_i \in C_j$ na posição k, o valor da precisão obtido para $\hat{C}_j^k$ na Equação (4.8) é zero, isto é, $\frac{|\hat{C}_j^k \cap C_j|}{|\hat{C}_j^k|} = 0$ [Manning08].

A definição original de *average precision* considera que existe somente uma categoria por posição no *ranking*, e a média das precisões é obtida dividindo o somatório das precisões pelo número de categorias pertinentes, $|C_j|$, pois $|C_j|$ precisões são computadas. Entretanto, nos *rankings* Denso, Padrão e Modificado mais de uma categoria pode pertencer a uma posição do *ranking*, e não necessariamente $|C_j|$ precisões são computadas. A nossa proposta de reformulação da métrica *average precision*, que chamamos de *average precision**, por outro lado, avalia a média das precisões computadas ao truncar o *ranking* de categorias para o documento de teste d_j após cada posição do *ranking* que tenha **pelo menos uma categoria** $c_i \in C_j$. A reformulação desta métrica é apresentada na Equação (4.9).

$$avg - precision_j* = \frac{1}{m}\sum_{k=1}^{m}\frac{\left|\hat{C}_j^k \cap C_j\right|}{\left|\hat{C}_j^k\right|} \quad (4.9)$$

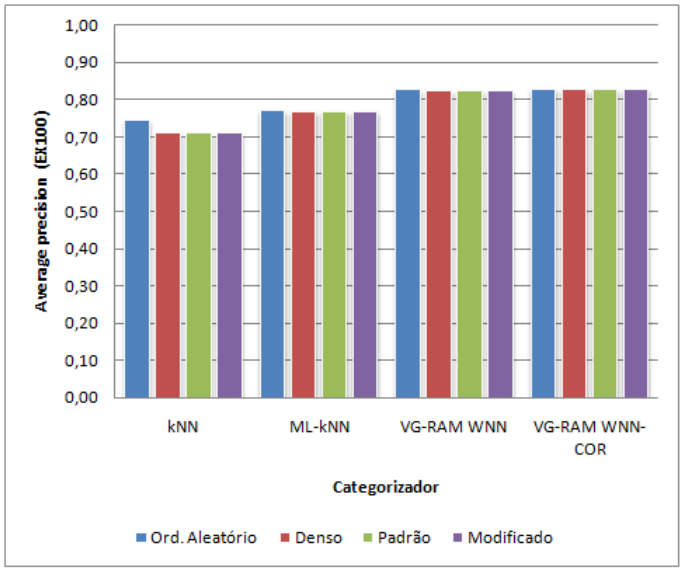
onde m é o número de posições no *ranking* que têm pelo menos uma categoria $c_i \in C_j$ para d_j, e $\hat{C}_j^k$ é o conjunto de categorias que vão do topo do *ranking* até a posição k do *ranking* que tem pelo menos uma categoria $c_i \in C_j$. Como antes, se existe uma categoria $c_i \in C_j$ na posição k e $f(d_j, c_i) = 0$, o valor da precisão obtido para $\hat{C}_j^k$ na Equação (4.9) é 0 (zero).

Note que, se existe mais de uma categoria pertencente a C_j na mesma posição k, o valor da precisão para $\hat{C}_j^k$ na Equação (4.9) é considerado somente uma vez, e por isso, a média de precisões é calculada dividindo-se o somatório das precisões por m, em vez de $|C_j|$. É importante observar que a métrica *average precision** é equivalente à definição original, se não existir empates, isto é, ela generaliza *average precision* para os casos de empates, penalizando os categorizadores que não são capazes de atribuir valores distintos de $f(d_j,.)$ para as categorias de interesse no *ranking*.

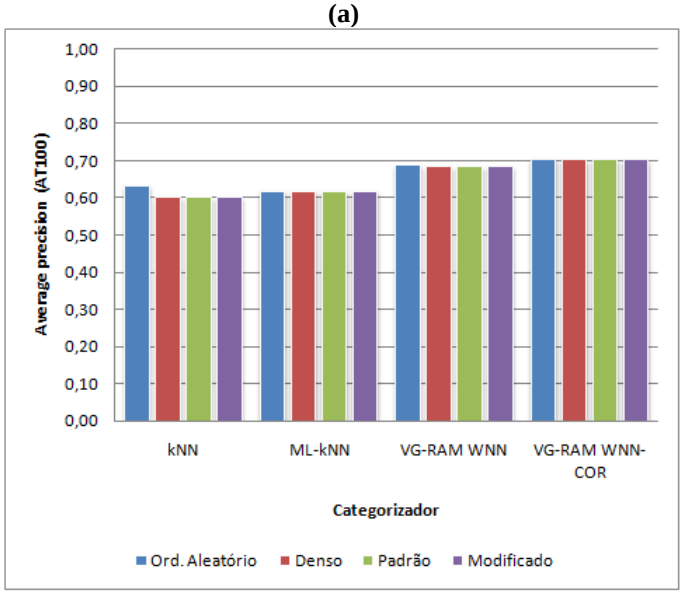
O desempenho global da métrica é calculado conforme a Equação (4.10). Quanto maior o valor de *average precision**, melhor o desempenho do sistema de categorização. O desempenho é perfeito quando $avg - precision* = 1$.

$$avg - precision* = \frac{1}{|Te|}\sum_{j=1}^{|Te|} avg - precision_j* \quad (4.10)$$

A Figura 4.4 mostra de forma gráfica o impacto de cada tipo de *ranking* na métrica *average precision** para a base EX100 (Figura 4.4(a)) e AT100 (Figura 4.4(b)). Esta figura segue o mesmo formato da Figura 4.1.



(a)



(b)

Figura 4.4 – Resultado da métrica *average precision para a base EX100, (a), e AT100, (b). Quanto maior, melhor.**

Conforme as barras do gráfico da Figura 4.4(a) mostram, o valor de *average precision** do categorizador *k NN* com a base EX100 é impactado pelo tipo de *ranking* empregado. O valor de *average precision** com o *ranking* Ordinal Aleatório é significativamente maior que com o *ranking* Denso, Padrão e Modificado (teste t pareado bicaudal com nível de significância 5%). O mesmo ocorre com os categorizadores *ML-k NN*, *VG-RAM WNN* e *VG-RAM WNN-COR* com esta base, e com todos os categorizadores com a base de dados AT100 (Figura 4.4(b)). O valor de *average precision** com os tipos de *ranking* Denso, Padrão e Modificado são os mesmos para ambas as bases, visto que a métrica *average precision** avalia o conjunto $\hat{C}_j^k$ das categorias do topo do *ranking* até a posição k do

ranking que tem pelo menos uma categoria $c_i \in C_j$, e o conjunto $\hat{C}_j^k$ nos *rankings* Denso, Padrão e Modificado são iguais, pois as posições vazias existentes nos *rankings* Padrão e Modificado são desconsideradas.

Como na Tabela 4.1, a Tabela 4.3 mostra a estatística t associada à comparação do desempenho dos categorizadores com o *ranking* Ordinal Aleatório, com o desempenho com os demais tipos de *ranking* em estudo segundo a métrica *average precision** para cada uma das bases de dados.

Tabela 4.3 – A estatística *t* da comparação do desempenho com o *ranking* Ordinal Aleatório, com o desempenho com os demais tipos de *ranking* segundo *average precision para as bases EX100 e AT100.**

Base de dados	Categorizador	Denso	Padrão	Modificado
EX100	kNN	*22,4633*	*22,4633*	*22,4633*
	ML-kNN	*7,0528*	*7,0528*	*7,0528*
	VG-RAM WNN	*8,4279*	*8,4279*	*8,4279*
	VG-RAM WNN-COR	*6,2347*	*6,2347*	*6,2347*
AT100	kNN	*24,5325*	*24,5325*	*24,5325*
	ML-kNN	*3,2380*	*3,2380*	*3,2380*
	VG-RAM WNN	*12,6988*	*12,6988*	*12,6988*
	VG-RAM WNN-COR	*6,2958*	*6,2958*	*6,2958*

Conforme a Tabela 4.3 mostra, o desempenho de todos os categorizadores segundo a métrica *average precision** é impactado pelos tipos de *ranking* Denso, Padrão e Modificado para as bases de dados EX100 e AT100, como discutido anteriormente. O valor de *average precision** destes categorizadores com o *ranking* Ordinal Aleatório é significativamente maior que com os *rankings* Denso e Padrão e Modificado.

Os resultados obtidos com os categorizadores *k NN*, *ML-k NN*, *VG-RAM WNN* e *VG-RAM WNN-COR* para a métrica *average precision** mostram que o *ranking* mais apropriado é o Denso, o Padrão, ou o Modificado. O *ranking* Ordinal Aleatório não é o mais apropriado, pois favoreceu os categorizadores que geraram *rankings* com empates entre as categorias de interesse no *ranking*.

4.1.5 R-precision

A métrica ***R-precision*** ($R-precision_j$) avalia a precisão computada com as $|C_j|$ categorias ordenadas no topo do *ranking* para o documento d_j. A formulação original de Baeza et al. [Baeza99] é apresentada na Equação (4.11).

$$R-precision_j = \frac{\left|\hat{C}_j^{|C_j|} \cap C_j\right|}{\left|\hat{C}_j^{|C_j|}\right|} \tag{4.11}$$

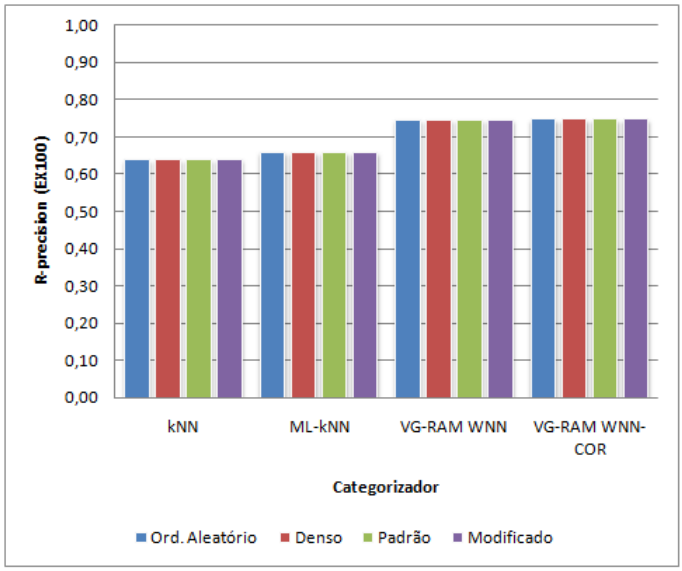
onde $\hat{C}_j^{|C_j|}$ é o conjunto de categorias $|C_j|$ ordenadas no topo do *ranking*. Entretanto, quando existem empates, o conjunto $|C_j|$ pode variar de acordo com o tipo de *ranking* utilizado. No caso do *ranking* Ordinal Aleatório, o conjunto $|C_j|$ refere-se a $|C_j|$ésima posição do *ranking*, enquanto que nos *rankings* Denso, Padrão e Modificado o conjunto $|C_j|$ pode-se referir a uma posição menor ou igual que a $|C_j|$ésima posição do *ranking*. Isso ocorre porque pode existir mais de uma categoria por posição nos *rankings* Denso, Padrão e Modificado .

Para obtermos o conjunto $\hat{C}_j^{|C_j|}$, incluímos todas as categorias por posição do *ranking* até completar as $|C_j|$ categorias de d_j, desconsiderando as posições vazias dos *rankings* Padrão e Modificado. Se na k ésima posição, que pode ser a $|C_j|$ésima, atingirmos $|C_j|$ e existir mais de uma categoria em k, inserimo-las no conjunto $\hat{C}_j^{|C_j|}$. Então, o conjunto $\hat{C}_j^{|C_j|}$ das $|C_j|$ categorias ordenadas no topo do *ranking* poder ser maior ou igual a $|C_j|$, isto é, $|\hat{C}_j^{|C_j|}| \geq |C_j|$. Note que categorias c_i das $|C_j|$ ordenadas no topo do *ranking* que possuem $f(d_j, c_i) = 0$ não são inseridas em $\hat{C}_j^{|C_j|}$. Neste caso, o conjunto $\hat{C}_j^{|C_j|}$ pode ser menor que $|C_j|$.

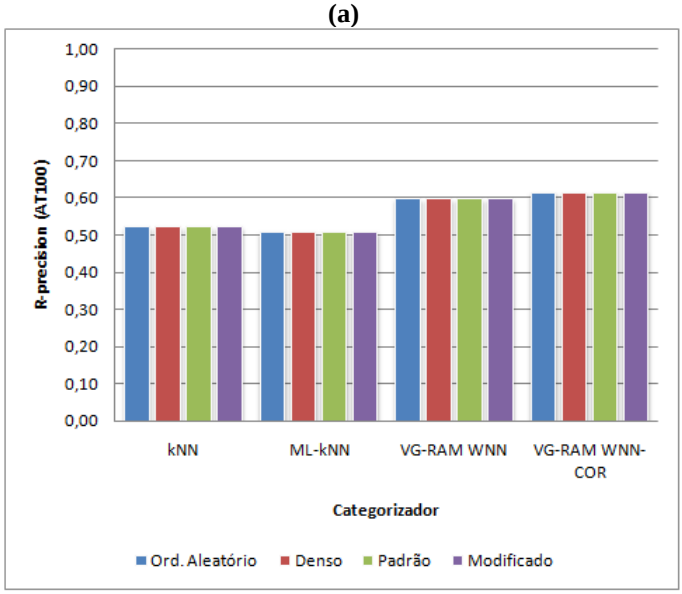
O desempenho global da métrica é calculado conforme Equação (4.12). Quanto maior o valor de $R-precision$, melhor o desempenho do categorizador. O desempenho é perfeito quando $R-precision = 1$.

$$R-precision = \frac{1}{|Te|}\sum_{j=1}^{|Te|} R-precision_j \tag{4.12}$$

A Figura 4.5 mostra de forma gráfica o impacto de cada tipo de *ranking* na métrica *R-precision* para a base EX100 (Figura 4.5(a)) e AT100 (Figura 4.5(b)). Esta figura segue o mesmo formato da Figura 4.1.



(a)



(b)

Figura 4.5 – Resultado da métrica *R-precision* para a base EX100, (a), e AT100, (b). Quanto maior, melhor.

Conforme as barras do gráfico da Figura 4.5(a) mostram, o valor de *R-precision* do categorizador *k NN* com a base EX100 não é impactado pelo tipo de *ranking* empregado. Ou seja, o valor de *R-precision* com o *ranking* Ordinal Aleatório não é significativamente diferente que com o *ranking* Denso, Padrão e Modificado (teste *t* pareado bicaudal com nível de significância 5%). O mesmo ocorre com os categorizadores *ML-k NN*, *VG-RAM WNN* e *VG-RAM WNN-COR* com esta base, e com todos os categorizadores com a base de dados AT100 (Figura 4.5(b)). O valor de *R-precision* com os tipos de *ranking* Denso, Padrão e Modificado são os mesmos para ambas as bases, visto que a métrica *R-precision* avalia o

conjunto $\hat{C}_j^{|C_j|}$ do *ranking* de categorias, e o conjunto $\hat{C}_j^{|C_j|}$ nos *rankings* Denso, Padrão e Modificado é igual, como explicado anteriormente.

Como na Tabela 4.1, a Tabela 4.4 mostra a estatística t associada à comparação do desempenho dos categorizadores com o *ranking* Ordinal Aleatório, com o desempenho com os demais tipos de *ranking* em estudo segundo a métrica *R-precision* para cada uma das bases de dados.

Tabela 4.4 – A estatística *t* da comparação do desempenho com o *ranking* Ordinal Aleatório, com o desempenho com os demais tipos de *ranking* segundo *R-precision* para as bases EX100 e AT100.

Base de dados	Categorizador	Denso	Padrão	Modificado
EX100	kNN	*-0,3577*	*-0,3577*	*-0,3577*
	ML-kNN	*1,9845*	*1,9845*	*1,9845*
	VG-RAM WNN	*1,5612*	*1,5612*	*1,5612*
	VG-RAM WNN-COR	*0,7982*	*0,7982*	*0,7982*
AT100	kNN	*-0,0711*	*-0,0711*	*-0,0711*
	ML-kNN	*0,2189*	*0,2189*	*0,2189*
	VG-RAM WNN	*0,4723*	*0,4723*	*0,4723*
	VG-RAM WNN-COR	*0,5481*	*0,5481*	*0,5481*

Como a Tabela 4.4 mostra, o desempenho de todos os categorizadores segundo a métrica *R-precision* não é impactado pelos tipos de *ranking* Denso, Padrão e Modificado para as bases de dados EX100 e AT100, como discutido anteriormente.

Os resultados obtidos com os categorizadores *k NN*, *ML-k NN*, *VG-RAM WNN* e *VG-RAM WNN-COR* para a métrica *R-precision* mostram que o desempenho dos categorizadores segundo esta métrica não é impactado pelos tipos de *ranking* Denso, Padrão, ou Modificado. Então, o *ranking* mais apropriado para a métrica *R-precision* é o Ordinal Aleatório, o Denso, o Padrão ou o Modificado.

4.2 Métricas de avaliação para conjuntos não-ordenados

As métricas de avaliação examinadas nesta subseção avaliam o conjunto de categorias preditas para o documento de teste d_j, $\hat{C}_j$, em vez de todo o *ranking*, como apresentado na Subseção 4.1. Em função disso, algum método para cortar (podar) o *ranking* de categorias derivado de $f(d_j, c_i)$ é necessário. Existem várias técnicas para determinar o limiar τ_i para cada categoria c_i [Yang01, Sebastiani02]. Contudo, como estamos somente interessados no

efeito dos tipos de *rankings* sobre as métricas usadas para avaliar as técnicas de categorização, avaliamos o desempenho dos categorizadores empregados sob uma política de corte perfeita, isto é, escolhemos a cardinalidade do conjunto de categorias preditas para d_j, $|\hat{C}_j|$, ser igual a $|C_j|$ (ou, quando existem empates, aproximadamente igual a $|C_j|$). Então, como foi feito para a métrica *R-precision*, derivamos $\hat{C}_j$ das $|C_j|$ categorias ordenadas no topo do *ranking* para d_j, e chamamos o conjunto obtido de $\hat{C}_j^{|C_j|}$.

Quando não existem empates, $\hat{C}_j^{|C_j|}$ contém simplesmente as $|C_j|$ categorias ordenadas no topo do *ranking*. Como em *R-precision*, se existem outras categorias na mesma posição do *ranking* da $|C_j|$ésima categoria, inserimo-las no conjunto $\hat{C}_j^{|C_j|}$. Então, quando existem outras categorias na mesma posição da $|C_j|$ésima categoria, $|\hat{C}_j^{|C_j|}| \geq |C_j|$. Também, categorias c_i no conjunto das $|C_j|$ categorias ordenadas no topo do *ranking* com $f(d_j, c_i) = 0$ não são inseridas em $\hat{C}_j^{|C_j|}$, e neste caso, o conjunto $\hat{C}_j^{|C_j|}$ pode ser menor que $|C_j|$.

Como as métricas apresentadas nesta subseção avaliam um conjunto não ordenado de categorias, as mesmas não precisam ser reformuladas para o caso de tratamento de empates com os *rankings* Denso, Padrão e Modificado. O conjunto $\hat{C}_j^{|C_j|}$ obtido para esses tipos de *ranking* é o mesmo, pois as posições vazias existentes nos tipos de *ranking* Padrão e Modificado são desconsideradas, como em *R-precision*. Então, o valor da métrica obtido com esses tipos de *ranking* é igual em cada categorizador tanto na base EX100 quanto na AT100.

4.2.1 Hamming loss

A métrica ***Hamming loss*** (*hamming-loss*$_j$) avalia quantas vezes o documento de teste d_j é categorizado erroneamente (isto é, uma categoria não pertinente ao documento é predita ou uma categoria pertinente ao documento não é predita), normalizada pelo número total de categorias. A formulação original de Schapire et al. [Schapire99] é apresentada na Equação (4.13).

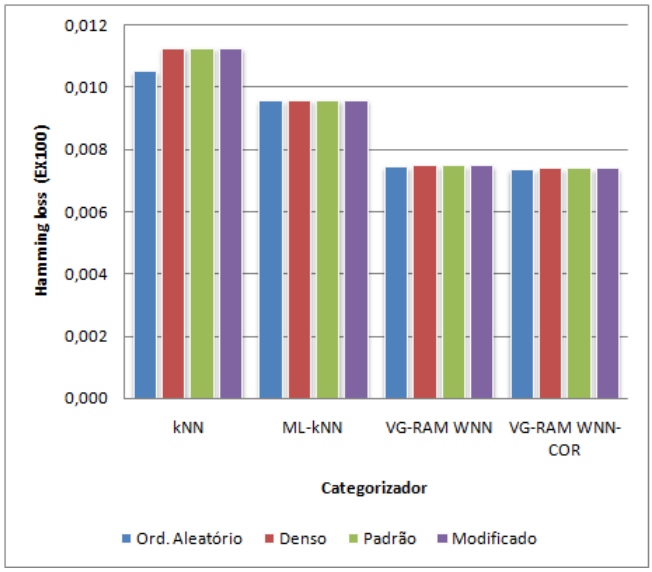
$$hamming\text{-}loss_j = \frac{\left|\hat{C}_j^{|C_j|} \Theta C_j\right|}{|C|} \tag{4.13}$$

onde Θ é a diferença simétrica entre o conjunto de categorias preditas, $\hat{C}_j^{|C_j|}$, e o conjunto de categorias pertinentes de d_j, C_j. O desempenho global é calculado conforme a Equação (4.14).

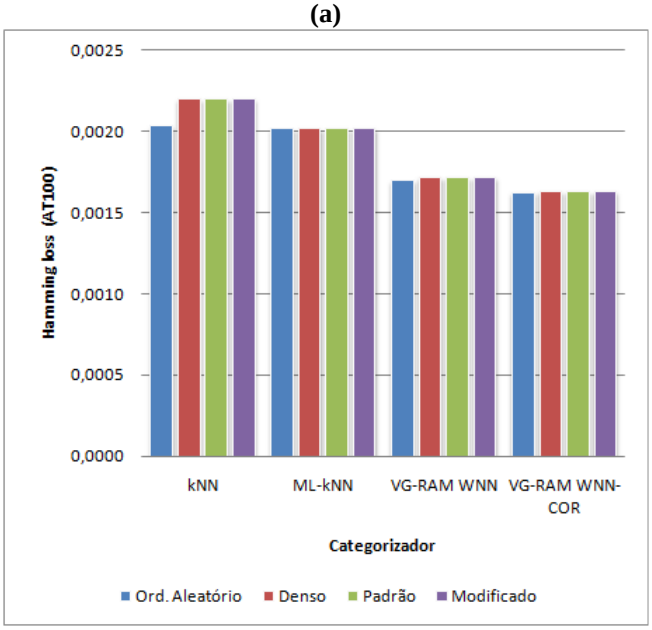
$$ham\min g - loss = \frac{1}{|Te|}\sum_{j=1}^{|Te|} ham\min g - loss_j \tag{4.14}$$

Quanto menor o valor de $ham\min g - loss$, melhor o desempenho do categorizador. O desempenho é perfeito quando $ham\min g - loss = 0$.

A Figura 4.6 mostra de forma gráfica o impacto de cada tipo de *ranking* na métrica *Hamming loss* para a base EX100 (Figura 4.6(a)) e AT100 (Figura 4.6(b)). Esta figura segue o mesmo formato da Figura 4.1.



(a)



(b)

Figura 4.6 – Resultado da métrica *Hamming loss* para a base EX100, (a), e AT100, (b). Quanto menor, melhor.

Como as barras do gráfico da Figura 4.6(a) mostram, o valor de *Hamming loss* do categorizador *k NN* com a base EX100 é impactado pelo tipo de *ranking* empregado. O valor de *Hamming loss* com o *ranking* Ordinal Aleatório é significativamente menor que com o *ranking* Denso, Padrão e Modificado (teste *t* pareado bicaudal com nível de significância 5%). O mesmo ocorre com os categorizadores *VG-RAM WNN* e *VG-RAM WNN-COR* com esta base, e com os categorizadores *k NN*, *VG-RAM WNN* e *VG-RAM WNN-COR* com a base de dados AT100 (Figura 4.6(b)).

A análise do desempenho do categorizador *ML-k NN* segundo a métrica *Hamming loss* mostra que o desempenho deste categorizador é significativamente afetado pelo tipo de

ranking para a base de dados EX100, apesar do conjunto de barras da Figura 4.6(a) associado a este categorizador não permita ver isso claramente. O valor de *Hamming loss* com o *ranking* Ordinal Aleatório é significativamente menor que com o *ranking* Denso, Padrão e Modificado, conforme mostra a Tabela 4.5, detalhada a seguir.

Como na Tabela 4.1, a Tabela 4.5 mostra a estatística t associada à comparação do desempenho dos categorizadores com o *ranking* Ordinal Aleatório, com o desempenho com os demais tipos de *ranking* em estudo segundo a métrica *Hamming loss* para cada uma das bases de dados.

Tabela 4.5 – A estatística *t* da comparação do desempenho com o *ranking* Ordinal Aleatório, com o desempenho com os demais tipos de *ranking* segundo *Hamming loss* para as bases EX100 e AT100.

Base de dados	Categorizador	Denso	Padrão	Modificado
EX100	kNN	*-11,8432*	*-11,8432*	*-11,8432*
	ML-kNN	*-2,3391*	*-2,3391*	*-2,3391*
	VG-RAM WNN	*-3,5699*	*-3,5699*	*-3,5699*
	VG-RAM WNN-COR	*-6,4397*	*-6,4397*	*-6,4397*
AT100	kNN	*-14,5173*	*-14,5173*	*-14,5173*
	ML-kNN	*-0,8258*	*-0,8258*	*-0,8258*
	VG-RAM WNN	*-7,2602*	*-7,2602*	*-7,2602*
	VG-RAM WNN-COR	*-4,6989*	*-4,6989*	*-4,6989*

De acordo com a Tabela 4.5, o desempenho do categorizador *ML-k NN* não é afetado pelo tipo de *ranking* para a base de dados AT100. Ou seja, o valor de *Hamming loss* deste categorizador com o *ranking* Ordinal Aleatório não é significativamente diferente que com os *rankings* Denso, Padrão e Modificado.

Os resultados obtidos com os categorizadores *k NN*, *ML-k NN*, *VG-RAM WNN* e *VG-RAM WNN-COR* para a métrica *Hamming loss* mostram que o *ranking* mais apropriado é o Denso, o Padrão, ou o Modificado. O *ranking* Ordinal Aleatório não é o mais apropriado, pois favoreceu os categorizadores que geraram *rankings* com empates entre as categorias de interesse no conjunto $\hat{C}_j^{|C_j|}$, com exceção do categorizador *ML-k NN* para a base de dados AT100.

4.2.2 Exact match

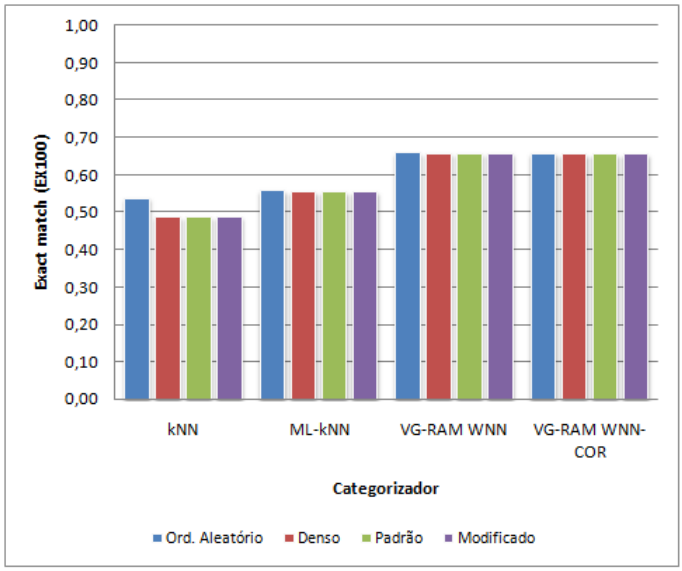
A métrica ***exact match*** ($exact-match_j$) avalia o quão freqüente todas e somente todas as categorias pertinentes estão presentes no conjunto de categorias preditas de d_j. A formulação original de Kazawa et al. [Kazawa05] é apresentada na Equação (4.15).

$$exact-macth_j = \begin{cases} 1 \;\; se \;\; \hat{C}_j^{|C_j|} = C_j \\ 0 \;\; caso \;\; contrário \end{cases} \tag{4.15}$$

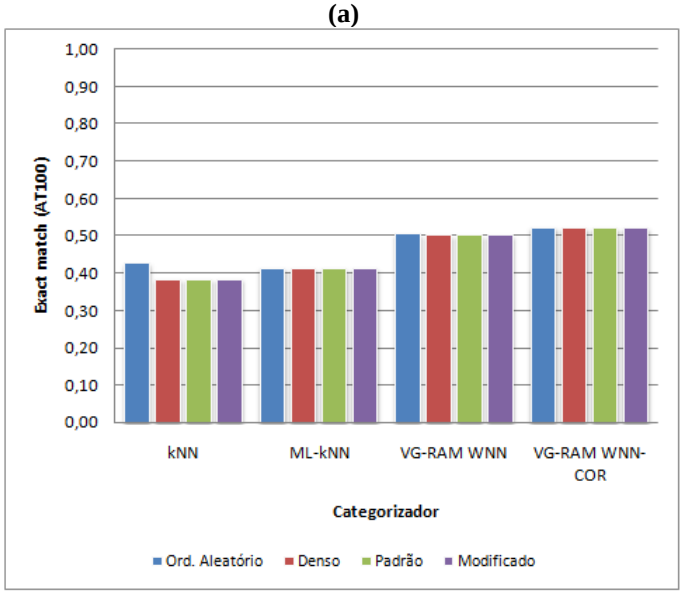
Se o conjunto $\hat{C}_j^{|C_j|}$ é igual ao conjunto C_j, $exact-macth_j = 1$, caso contrário, $exact-macth_j = 0$. O desempenho global é obtido conforme Equação (4.16). Quanto maior o valor de *exact match*, melhor o desempenho do categorizador. O desempenho é perfeito quando $exact-match = 1$.

$$exact-match = \frac{1}{|Te|}\sum_{j=1}^{|Te|} exact-match_j \tag{4.16}$$

A Figura 4.7 mostra de forma gráfica o impacto de cada tipo de *ranking* na métrica *exact match* para a base EX100 (Figura 4.7(a)) e AT100 (Figura 4.7(b)). Esta figura segue o mesmo formato da Figura 4.1.



(a)



(b)

Figura 4.7 – Resultado da métrica *exact match* para a base EX100, (a), e AT100, (b).Quanto maior, melhor.

Conforme as barras do gráfico da Figura 4.7(a) mostram, o valor de *exact match* do categorizador *k NN* com a base EX100 é impactado pelo tipo de *ranking* empregado. O valor de *exact match* com o *ranking* Ordinal Aleatório é significativamente maior que com o *ranking* Denso, Padrão e Modificado (teste *t* pareado bicaudal com nível de significância 5%). O mesmo ocorre com os categorizadores *ML-k NN* e *VG-RAM WNN* com esta base, e com os categorizadores *k NN* e *VG-RAM WNN* com a base de dados AT100 (Figura 4.7(b)).

A análise do desempenho do categorizador *ML-k NN* segundo a métrica *exact match* mostra que o desempenho deste categorizador não é afetado pelo tipo de *ranking* para a base de dados AT100 (Figura 4.7(b)). Ou seja, o valor de *exact match* deste categorizador com o

ranking Ordinal Aleatório não é significativamente diferente que com os *rankings* Denso, Padrão e Modificado, conforme mostra a Tabela 4.6, detalhada a seguir.

Como na Tabela 4.1, a Tabela 4.6 apresenta a estatística t associada à comparação do desempenho dos categorizadores com o *ranking* Ordinal Aleatório, com o desempenho com os demais tipos de *ranking* em estudo segundo a métrica *exact match* para cada uma das bases de dados.

Tabela 4.6 – A estatística *t* da comparação do desempenho com o *ranking* Ordinal Aleatório, com o desempenho com os demais tipos de *ranking* segundo *exact match* para as bases EX100 e AT100.

Base de dados	Categorizador	Denso	Padrão	Modificado
EX100	kNN	*15,8363*	*15,8363*	*15,8363*
	ML-kNN	*4,6694*	*4,6694*	*4,6694*
	VG-RAM WNN	*3,8569*	*3,8569*	*3,8569*
	VG-RAM WNN-COR	*4,3336*	*4,3336*	*4,3336*
AT100	kNN	*16,6559*	*16,6559*	*16,6559*
	ML-kNN	*1,9219*	*1,9219*	*1,9219*
	VG-RAM WNN	*8,7128*	*8,7128*	*8,7128*
	VG-RAM WNN-COR	*2,7633*	*2,7633*	*2,7633*

Como a Figura 4.7 mostra, o impacto do tipo de *ranking* no desempenho do categorizador *VG-RAM WNN-COR* é similar àquele observado no categorizador *ML-k NN*, muito embora não seja idêntico. De acordo com a Tabela 4.6, o desempenho do categorizador *VG-RAM WNN-COR* é significativamente maior que com o *ranking* Denso, Padrão e Modificado para a base de dados AT100.

Os resultados obtidos com os categorizadores *k NN*, *ML-k NN*, *VG-RAM WNN* e *VG-RAM WNN-COR* para a métrica *exact match* mostram que o *ranking* mais apropriado é o Denso, o Padrão, ou o Modificado. O *ranking* Ordinal Aleatório não é o mais apropriado, pois favoreceu os categorizadores que geraram *rankings* com empates entre as categorias de interesse no conjunto $\hat{C}_j^{|C_j|}$, com exceção do categorizador *ML-k NN* para a base de dados AT100.

4.2.3 Precisão (*precision*) orientada à categoria

A métrica ***precisão (precision)* orientada à categoria** ($precision_i^c$) avalia a fração de documentos de teste categorizados sob a categoria c_i que são verdadeiramente associados a c_i. A formulação é apresentada na Equação (4.17).

$$precision_i^c = \frac{\left|\hat{C}_j^{|C_j|} \cap C_j\right|}{|\hat{C}_j^{|C_j|}|} \tag{4.17}$$

A métrica *precision* orientada à categoria também pode ser computada utilizando a tabela de contingência da categoria c_i (Tabela 4.7), de acordo com a Equação (4.18).

$$precision_i^c = \frac{TP_i}{TP_i + FP_i} \tag{4.18}$$

onde FP_i (falsos positivos para c_i) é o número de documentos de teste que foram incorretamente categorizados sob c_i; TN_i (verdadeiros negativos para c_i) é o número de documentos de teste que foram corretamente não categorizados sob c_i; TP_i (verdadeiros positivos para c_i) é o número de documentos de teste que foram corretamente categorizados sob c_i; e FN_i (falsos negativos para c_i) é o número de documentos de teste que foram incorretamente não categorizados sob c_i.

Tabela 4.7 – Tabela de contingência da categoria c_i.

Categoria c_i		**Julgamentos do especialista**	
		SIM	NÃO
Julgamentos do categorizador	SIM	TP_i	FP_i
	NÃO	FN_i	TN_i

O desempenho global de *precision* orientada à categoria pode ser computado pelo método *macroaveraging* ($macro - precision^c$) e *microaveraging* ($micro - precision^c$), Equação (4.19) e Equação (4.20), respectivamente [Sebatiani2002]. O método *macroaveraging* reporta o desempenho global sobre a soma dos resultados de $precision_i^c$ (Equação (4.19)), e o *microaveraging* sobre a soma das decisões individuais em termos da tabela de contingência, $TP_i \big/ (TP_i + FP_i)$ (Equação (4.20)), para cada categoria c_i.

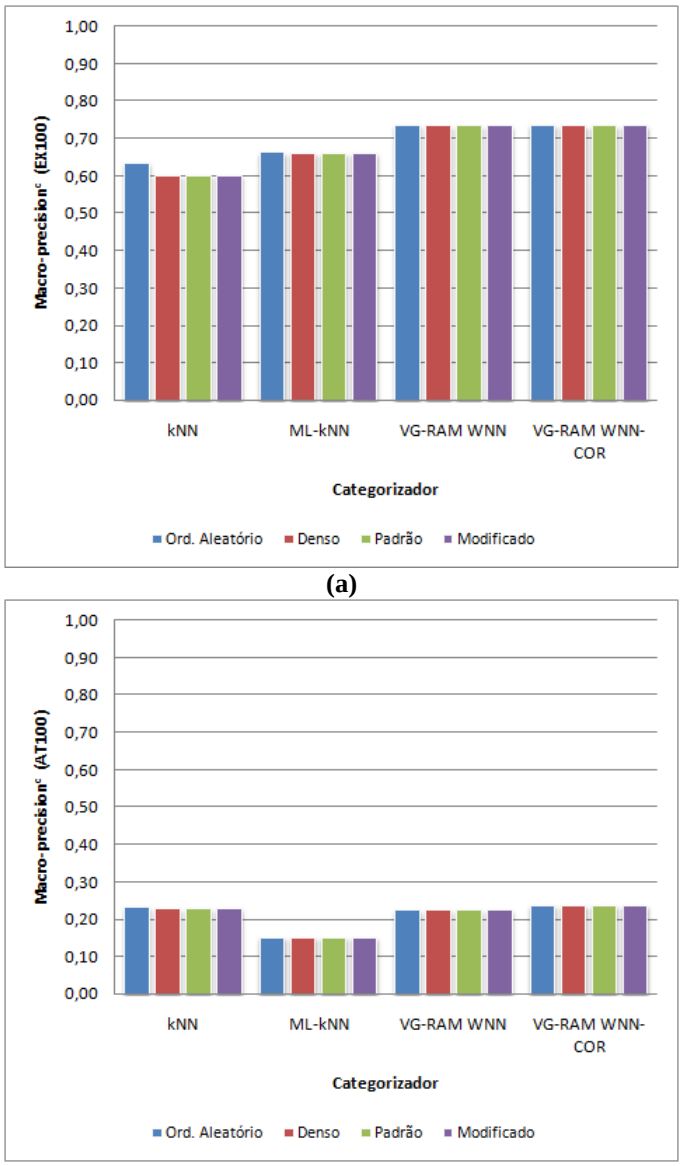
$$macro - precision^c = \frac{\sum_{i=1}^{|C|} precision_i^c}{|C|} \tag{4.19}$$

$$micro-precision^{c} = \frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|} (TP_i + FP_i)} \quad (4.20)$$

Os métodos *macroaveraging* e *microaveraging* podem dar resultados bastante diferentes, especialmente se as generalidades das categorias são desiguais [Manning08; Sebastiani02]. A habilidade de um categorizador de se comportar bem mediante categorias com baixa generalidade é evidenciada muito mais por *macroaveraging* e do que por *microaveraging*. O método *macroaveraging* dá peso igual para cada categoria, enquanto *microaveraging* dá peso igual para cada decisão de categorização [Manning08]. Desta forma, categorias com alta generalidade dominam aquelas com baixa generalidade em *microaveraging*.

Quanto maior o valor de $macro-precision^{c}$ e $micro-precision^{c}$ melhor o desempenho do categorizador. O desempenho é perfeito quando $macro-precision^{c}=1$ e $micro-precision^{c}=1$.

A Figura 4.8 mostra de forma gráfica o impacto de cada tipo de *ranking* na métrica $macro-precision^{c}$ para a base EX100 (Figura 4.8(a)) e AT100 (Figura 4.8(b)). Esta figura segue o mesmo formato da Figura 4.1.



(a)

(b)

Figura 4.8 – Resultado da métrica $macro-precision^c$ para a base EX100, (a), e AT100, (b). Quanto maior, melhor

Conforme as barras do gráfico da Figura 4.8(a) mostram, o valor de $macro-precision^c$ do categorizador $k\,NN$ com a base EX100 é impactado pelo tipo de *ranking* empregado. O valor de $macro-precision^c$ com o *ranking* Ordinal Aleatório é significativamente maior que com o *ranking* Denso, Padrão e Modificado (teste t pareado bicaudal com nível de significância 5%). O mesmo ocorre com a base de dados AT100 (Figura 4.8(b)).

A análise do desempenho do categorizador $ML\text{-}k\,NN$ segundo a métrica $macro-precision^c$ mostra que o desempenho deste categorizador também é afetado pelo tipo

de *ranking* para a base de dados EX100 (Figura 4.8(a)). O valor de $macro-precision^c$ com o *ranking* Ordinal Aleatório é significativamente maior que com o *ranking* Denso, Padrão e Modificado, conforme mostra a Tabela 4.8, detalhada a seguir.

Como na Tabela 4.1, a Tabela 4.8 apresenta a estatística t associada à comparação do desempenho dos categorizadores com o *ranking* Ordinal Aleatório, com o desempenho com os demais tipos de *ranking* em estudo segundo a métrica $macro-precision^c$ para cada uma das bases de dados.

Tabela 4.8 – A estatística *t* da comparação do desempenho com o *ranking* Ordinal Aleatório, com o desempenho com os demais tipos de *ranking* segundo $macro-precision^c$ para as bases EX100 e AT100.

Base de dados	Categorizador	Denso	Padrão	Modificado
EX100	kNN	*17,1117*	*17,1117*	*17,1117*
	ML-kNN	*2,2666*	*2,2666*	*2,2666*
	VG-RAM WNN	*3,8837*	*3,8837*	*3,8837*
	VG-RAM WNN-COR	*1,8532*	*1,8532*	*1,8532*
AT100	kNN	*3,4943*	*3,4943*	*3,4943*
	ML-kNN	*0,9781*	*0,9781*	*0,9781*
	VG-RAM WNN	*-0,5370*	*-0,5370*	*-0,5370*
	VG-RAM WNN-COR	*0,9503*	*0,9503*	*0,9503*

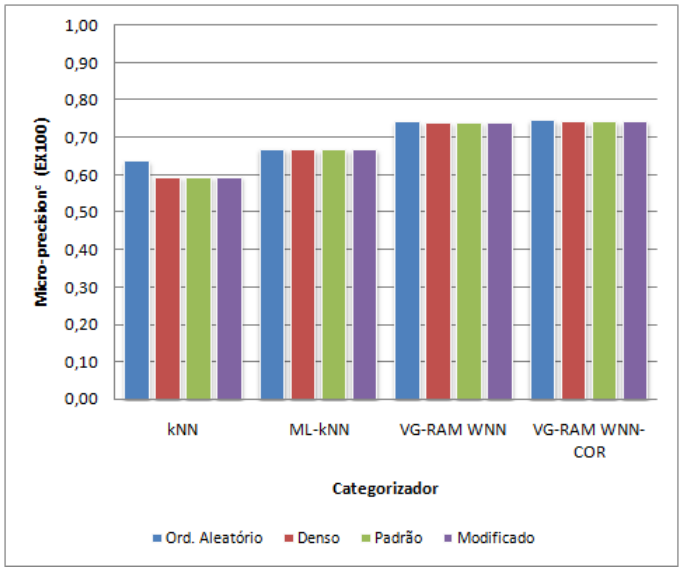
De acordo com a Tabela 4.8, o desempenho do categorizador *ML-k NN* não é afetado pelo tipo de *ranking* para a base de dados AT100. Ou seja, o valor de $macro-precision^c$ deste categorizador com o *ranking* Ordinal Aleatório não é significativamente diferente que com os *rankings* Denso, Padrão e Modificado.

Como a Figura 4.8 mostra, o impacto do tipo de *ranking* no desempenho dos categorizadores *VG-RAM WNN* e *VG-RAM WNN-COR* é similar àquele observado no categorizador *ML-k NN*, muito embora não seja idêntico. De acordo com a Tabela 4.8, o desempenho do categorizador *VG-RAM WNN* segundo a métrica $macro-precision^c$ é significativamente afetado pelos tipos de *ranking* para a base de dados EX100. Entretanto, o desempenho deste categorizador não é impactado pelos tipos de *ranking* em estudo para a base de dados AT100. Na análise de desempenho do categorizador *VG-RAM WNN-COR* observa-se que o desempenho deste categorizador não é afetado pelos tipos de *ranking* para as bases de dados EX100 e AT100.

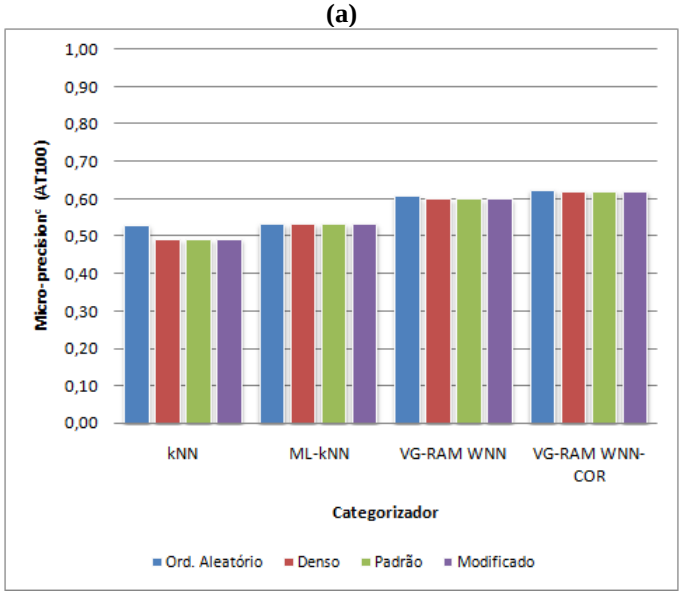
Os resultados obtidos com os categorizadores *k NN*, *ML-k NN*, *VG-RAM WNN* e *VG-RAM WNN-COR* para a métrica $macro-precision^c$ mostram que o *ranking* mais apropriado é o Denso, o Padrão, ou o Modificado. O *ranking* Ordinal Aleatório não é o mais apropriado,

pois favoreceu os categorizadores que geraram *rankings* com empates entre as categorias de interesse no conjunto $\hat{C}_j^{|C_j|}$.

A Figura 4.9 mostra de forma gráfica o impacto de cada tipo de *ranking* na métrica $micro-precision^c$ para a base EX100 (Figura 4.9(a)) e AT100 (Figura 4.9(b)). Esta figura segue o mesmo formato da Figura 4.1.



(a)



(b)

Figura 4.9 – Resultado da métrica $micro-precision^c$ para a base EX100, (a), e AT100, (b). Quanto maior, melhor.

Conforme as barras do gráfico da Figura 4.9(a) mostram, o valor de $micro-precision^c$ do categorizador $k\,NN$ com a base EX100 é impactado pelo tipo de

ranking empregado. O valor de $micro-precision^c$ com o *ranking* Ordinal Aleatório é significativamente maior que com o *ranking* Denso, Padrão e Modificado (teste t pareado bicaudal com nível de significância 5%). O mesmo ocorre com os categorizadores *VG-RAM WNN* e *VG-RAM WNN-COR* com esta base, e com os categorizadores k *NN*, *VG-RAM WNN* e *VG-RAM WNN-COR* com a base de dados AT100 (Figura 4.9(b)).

A análise do desempenho do categorizador *ML-*k *NN* segundo a métrica $micro-precision^c$ mostra que o desempenho deste categorizador também é afetado pelo tipo de *ranking* para a base de dados EX100 (Figura 4.9(a)). O valor de $micro-precision^c$ com o *ranking* Ordinal Aleatório é significativamente maior que com o *ranking* Denso, Padrão e Modificado, conforme mostra a Tabela 4.9, detalhada a seguir.

Como na Tabela 4.1, a Tabela 4.9 apresenta a estatística t associada à comparação do desempenho dos categorizadores com o *ranking* Ordinal Aleatório, com o desempenho com os demais tipos de *ranking* em estudo segundo a métrica $micro-precision^c$ para cada uma das bases de dados.

Tabela 4.9 – A estatística *t* da comparação do desempenho com o *ranking* Ordinal Aleatório, com o desempenho com os demais tipos de *ranking* segundo $micro-precision^c$ para as bases EX100 e AT100.

Base de dados	Categorizador	Denso	Padrão	Modificado
EX100	kNN	*20,8092*	*20,8092*	*20,8092*
	ML-kNN	*3,6645*	*3,6645*	*3,6645*
	VG-RAM WNN	*5,9509*	*5,9509*	*5,9509*
	VG-RAM WNN-COR	*11,6648*	*11,6648*	*11,6648*
AT100	kNN	*16,0190*	*16,0190*	*16,0190*
	ML-kNN	*0,9505*	*0,9505*	*0,9505*
	VG-RAM WNN	*10,2075*	*10,2075*	*10,2075*
	VG-RAM WNN-COR	*5,9143*	*5,9143*	*5,9143*

Como a Tabela 4.9 mostra, o desempenho do categorizador *ML-*k *NN* não é afetado pelo tipo de *ranking* para a base de dados AT100. Ou seja, o valor de $micro-precision^c$ deste categorizador com o *ranking* Ordinal Aleatório não é significativamente diferente que com os *rankings* Denso, Padrão e Modificado.

Os resultados obtidos com os categorizadores k *NN*, *ML-*k *NN*, *VG-RAM WNN* e *VG-RAM WNN-COR* para a métrica $micro-precision^c$ mostram que o *ranking* mais apropriado é o Denso, o Padrão, ou o Modificado. O *ranking* Ordinal Aleatório não é o mais apropriado, pois favoreceu os categorizadores que geraram *rankings* com empates entre as categorias de

interesse no conjunto $\hat{C}_j^{|C_j|}$, com exceção do categorizador *ML-k NN* para bases de dados AT100.

4.2.4 Revocação (*recall*) orientada à categoria

A métrica **revocação *(recall)* orientada à categoria (** $recall_i^c$ **)** avalia a fração de documentos de teste verdadeiramente associados com a categoria c_i que são categorizados sob c_i. A formulação original é apresentada na Equação (4.21).

$$recall_i^c = \frac{\left|\hat{C}_j^{|C_j|} \cap C_j\right|}{|C_j|} \tag{4.21}$$

O valor de $recall_i^c$ também ser computado em termos da tabela de contingência da categoria c_i, Tabela 4.7, conforme Equação (4.22).

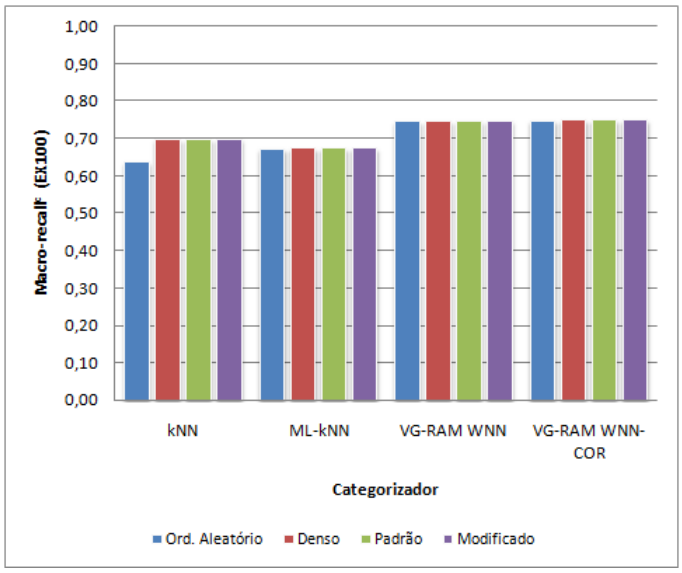
$$recall_i^c = \frac{TP_i}{TP_i + FN_i} \tag{4.22}$$

O desempenho global de recall orientada à categoria é calculado pelos métodos $macro - recall^c$ e $micro - recall^c$, Equação (4.23) e Equação (4.24), respectivamente. Quanto maior o valor de $macro - recall^c$ e $micro - recall^c$, melhor o desempenho do categorizador. O desempenho é perfeito quando $macro - recall^c = 1$ e $micro - recall^c = 1$.

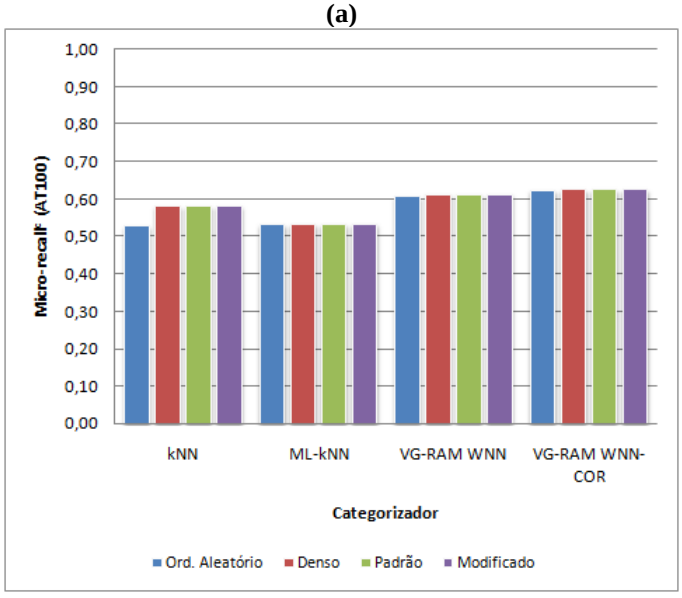
$$macro - recall^c = \frac{\sum_{i=1}^{|C|} recall_i^c}{|C|} \tag{4.23}$$

$$micro - recall^c = \frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|} (TP_i + FN_i)} \tag{4.24}$$

A Figura 4.10 mostra de forma gráfica o impacto de cada tipo de *ranking* na métrica $macro-recall^c$ para a base EX100 (Figura 4.10(a)) e AT100 (Figura 4.10(b)). Esta figura segue o mesmo formato da Figura 4.1.



(a)



(b)

Figura 4.10 – Resultado da métrica $macro-recall^c$ para a base EX100, (a), e AT100, (b). Quanto maior, melhor.

Conforme as barras do gráfico da Figura 4.10(a) mostram, o valor de $macro-recall^c$ do categorizador $k\,NN$ com a base EX100 é impactado pelo tipo de *ranking* empregado. O valor de $macro-recall^c$ com o *ranking* Ordinal Aleatório é significativamente menor que com o *ranking* Denso, Padrão e Modificado (teste t pareado bicaudal com nível de

significância 5%). O mesmo ocorre com os categorizadores *ML-k NN* e *VG-RAM WNN-COR* com esta base, e com todos os categorizadores com a base de dados AT100 (Figura 4.10(b)).

A análise do desempenho do categorizador *VG-RAM WNN* segundo a métrica $macro-recall^c$ mostra que o desempenho deste categorizador também é afetado pelo tipo de *ranking* para a base de dados EX100, apesar do conjunto de barras da Figura 4.10(a) associado a este categorizador não permita ver isso claramente. O valor de $macro-recall^c$ deste categorizador com o *ranking* Ordinal Aleatório significativamente menor que com os *rankings* Denso, Padrão e Modificado, conforme mostra a Tabela 4.10, detalhada a seguir.

Como na Tabela 4.1, a Tabela 4.10 apresenta a estatística t associada à comparação do desempenho dos categorizadores com o *ranking* Ordinal Aleatório, com o desempenho com os demais tipos de *ranking* em estudo segundo a métrica $macro-recall^c$ para cada uma das bases de dados.

Tabela 4.10 – A estatística *t* da comparação do desempenho com o *ranking* Ordinal Aleatório, com o desempenho com os demais tipos de *ranking* segundo $macro-recall^c$ para as bases EX100 e AT100.

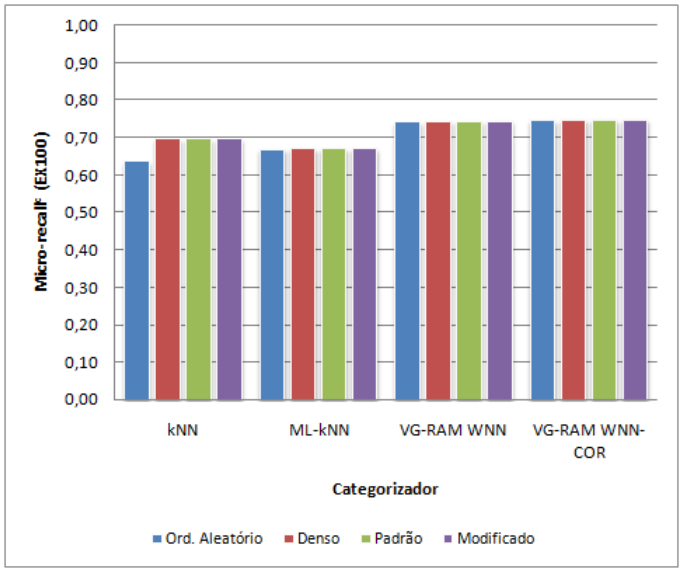
Base de dados	Categorizador	Denso	Padrão	Modificado
EX100	kNN	*-26,7825*	*-26,7825*	*-26,7825*
	ML-kNN	*-5,2226*	*-5,2226*	*-5,2226*
	VG-RAM WNN	*-3,1193*	*-3,1193*	*-3,1193*
	VG-RAM WNN-COR	*-4,8627*	*-4,8627*	*-4,8627*
AT100	kNN	*-16,5241*	*-16,5241*	*-16,5241*
	ML-kNN	*-2,8150*	*-2,8150*	*-2,8150*
	VG-RAM WNN	*-7,9144*	*-7,9144*	*-7,9144*
	VG-RAM WNN-COR	*-3,5318*	*-3,5318*	*-3,5318*

De acordo com a Tabela 4.10, o desempenho de todos os categorizadores segundo a métrica $macro-recall^c$ é impactado pelos tipos de *ranking* Denso, Padrão e Modificado para as bases de dados EX100 e AT100. O valor de $macro-recall^c$ destes categorizadores com o *ranking* Ordinal Aleatório é significativamente menor que com os *rankings* Denso, Padrão e Modificado.

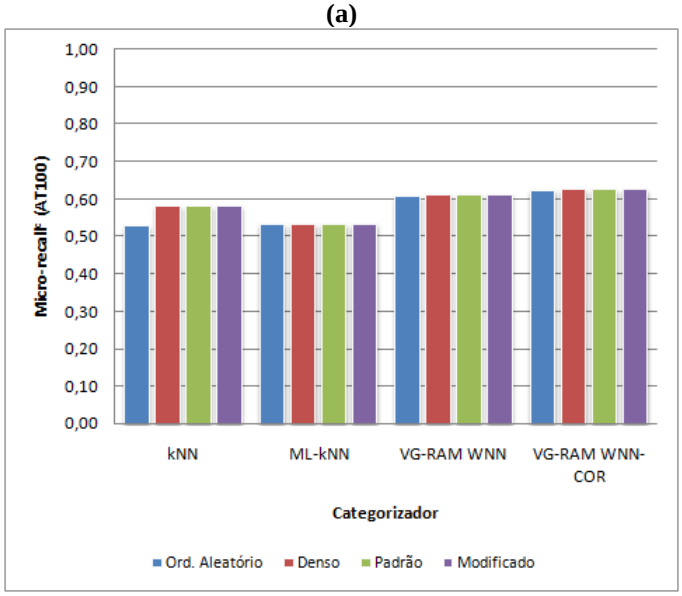
Aparentemente, os resultados obtidos com os categorizadores *k NN, ML-k NN, VG-RAM WNN* e *VG-RAM WNN-COR* para a métrica $macro-recall^c$ mostram que o *ranking* mais apropriado é o Ordinal Aleatório. Contudo, este *ranking* não avalia as categorias que estão empatadas com a $|C_j|$ésima categoria no *ranking*, isto é, as categorias empatadas com a $|C_j|$ésima categoria não são inseridas no conjunto $\hat{C}_j^{|C_j|}$. Com isso, a cardinalidade do conjunto $\hat{C}_j^{|C_j|}$ (numerador da Equação (4.21)) com o *ranking* Ordinal Aleatório é menor do

que a cardinalidade do conjunto $\hat{C}_j^{|C_j|}$ com os *rankings* Denso, Padrão e Modificado. Assim, o valor da métrica $macro-recall^c$ com o *ranking* Ordinal Aleatório é menor do que com os *rankings* Denso, Padrão e Modificado, como mostrado anteriormente. Então, os *rankings* Denso, Padrão e Modificado são os mais apropriados no caso da métrica $macro-recall^c$, pois avaliam o desempenho dos categorizadores de acordo com a política de corte adotada neste trabalho, e não apenas com uma das categorias empatadas com a $|C_j|$ésima categoria no *ranking*, como no *ranking* Ordinal Aleatório.

A Figura 4.11 mostra de forma gráfica o impacto de cada tipo de *ranking* na métrica $micro-recall^c$ para a base EX100 (Figura 4.11(a)) e AT100 (Figura 4.11(b)). Esta figura segue o mesmo formato da Figura 4.1.



(a)



(b)

Figura 4.11 – Resultado da métrica $micro-recall^c$ para a base EX100, (a), e AT100, (b). Quanto maior, melhor

O desempenho dos categorizadores *k NN*, *ML-k NN*, *VG-RAM WNN* e *VG-RAM WNN-COR* segundo a métrica $micro-recall^c$ é impactado pelos tipos de *ranking* em estudo para as duas bases de dados, de forma similar àquele observado na métrica $macro-recall^c$, conforme mostra a Tabela 4.11.

A Tabela 4.11 mostra a estatística t associada à comparação do desempenho dos categorizadores com o *ranking* Ordinal Aleatório, com o desempenho, com os demais tipos de *ranking* em estudo segundo a métrica $micro-recall^c$ para cada uma das bases de dados.

Tabela 4.11 – A estatística *t* da comparação do desempenho com o *ranking* Ordinal Aleatório, com o desempenho com os demais tipos de *ranking* segundo $micro-recall^c$ para as bases EX100 e AT100.

Base de dados	Categorizador	Denso	Padrão	Modificado
EX100	kNN	*-32,2633*	*-32,2633*	*-32,2633*
	ML-kNN	*-6,3973*	*-6,3973*	*-6,3973*
	VG-RAM WNN	*-3,4920*	*-3,4920*	*-3,4920*
	VG-RAM WNN-COR	*-5,3933*	*-5,3933*	*-5,3933*
AT100	kNN	*-28,0168*	*-28,0168*	*-28,0168*
	ML-kNN	*-2,8868*	*-2,8868*	*-2,8868*
	VG-RAM WNN	*-9,7551*	*-9,7551*	*-9,7551*
	VG-RAM WNN-COR	*-5,8698*	*-5,8698*	*-5,8698*

De acordo com a Tabela 4.11, o desempenho de todos os categorizadores segundo a métrica $micro-recall^c$ é impactado pelos tipos de *ranking* Denso, Padrão e Modificado para as bases de dados EX100 e AT100. O valor de $micro-recall^c$ destes categorizadores com o *ranking* Ordinal Aleatório é significativamente menor que com os *rankings* Denso, Padrão e Modificado.

Os resultados obtidos com os categorizadores *k NN*, *ML-k NN*, *VG-RAM WNN* e *VG-RAM WNN-COR* para a métrica $micro-recall^c$ mostram que o *ranking* mais apropriado é o Denso, o Padrão, ou o Modificado. O *ranking* Ordinal Aleatório não é o mais apropriado, conforme discutido anteriormente para a métrica $macro-recall^c$.

4.2.5 F_β orientada à categoria

A métrica F_β **orientada à categoria (**$F_{\beta_i}^c$**)** avalia a média harmônica ponderada de $precision_i^c$ e $recall_i^c$**.** A formulação original de Rijsbergen [Rijsbergen79] é mostrada na Equação (4.25).

$$F_{\beta_i}^{\ c} = \frac{(\beta^2+1) * precision_i^c * recall_i^c}{\beta^2 * precision_i^c + recall_i^c} \tag{4.25}$$

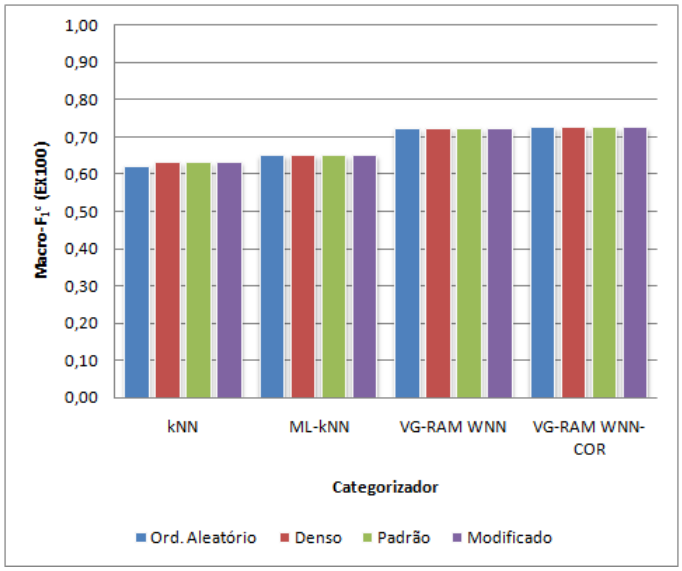
Na Equação (4.25), β pode ser visto como o grau relativo de importância atribuído para $precision_i^c$ e $recall_i^c$ [Sebastiani02]. Se $\beta = 0$, $F_{\beta_i}^c$ coincide com $precision_i^c$; $\beta = +\infty$, $F_{\beta_i}^c$ coincide com $recall_i^c$. Neste trabalho um valor de $\beta = 1$ é utilizado, atribuindo importância igual para $precision_i^c$ e $recall_i^c$. O desempenho global de F_1^c pode ser

computado tanto por $macro - F_1^c$ (Equação (4.26)) quanto $micro - F_1^c$ (Equação (4.27)). Quanto maior o valor de $macro - F_1^c$ e $micro - F_1^c$, melhor o desempenho do categorizador. O desempenho é perfeito quando $macro - F_1^c = 1$ e $micro - F_1^c = 1$.

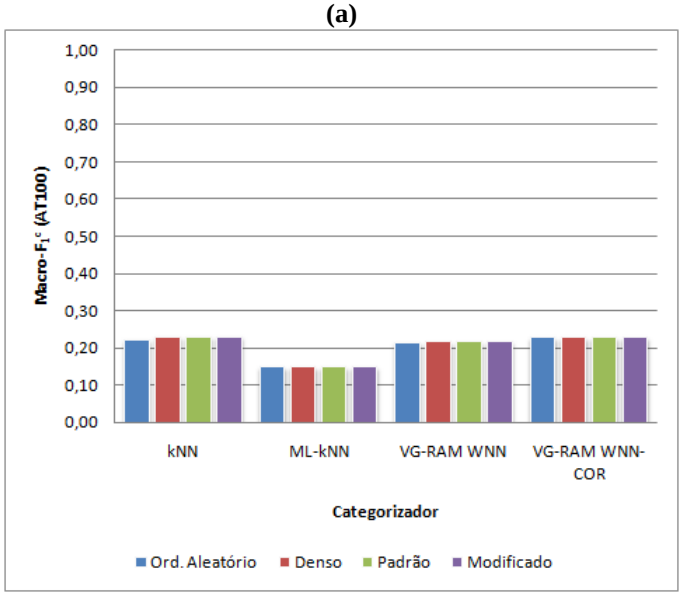
$$macro - F_1^c = \frac{1}{|C|}\sum_{i=1}^{C} F_{1i}^c \tag{4.26}$$

$$micro - F_1^c = \frac{2 * micro - precision_i^c * micro - recall_i^c}{micro - precision_i^c + micro - recall_i^c} \tag{4.27}$$

A Figura 4.12 mostra de forma gráfica o impacto de cada tipo de *ranking* na métrica $macro - F_1^c$ para a base EX100 (Figura 4.12(a)) e AT100 (Figura 4.12(b)). Esta figura segue o mesmo formato da Figura 4.1.



(a)



(b)

Figura 4.12 – Resultado da métrica $macro - F_1^c$ para a base EX100, (a), e AT100, (b). Quanto maior, melhor.

Conforme as barras do gráfico da Figura 4.12(a) mostram, o valor de $macro - F_1^c$ do categorizador $k\,NN$ com a base EX100 é impactado pelo tipo de *ranking* empregado. O valor de $macro - F_1^c$ com o *ranking* Ordinal Aleatório é significativamente menor que com o *ranking* Denso, Padrão e Modificado (teste t pareado bicaudal com nível de significância 5%). E o mesmo ocorre para a base de dados AT100 (Figura 4.12(b)).

A análise do desempenho do categorizador *ML-k NN* segundo a métrica $macro - F_1^c$ mostra que o desempenho deste categorizador não é afetado pelo tipo de *ranking* para a base de dados EX100 (Figura 4.12(a)). O mesmo ocorre com os categorizadores *VG-RAM WNN* e

VG-RAM WNN-COR com esta base, e com os categorizadores *ML-k NN* e *VG-RAM WNN-COR* com a base de dados AT100 (Figura 4.12(b)).

Como na Tabela 4.1, a Tabela 4.12 apresenta a estatística *t* associada à comparação do desempenho dos categorizadores com o *ranking* Ordinal Aleatório, com o desempenho com os demais tipos de *ranking* em estudo segundo a métrica $macro - F_1^c$ para cada uma das bases de dados.

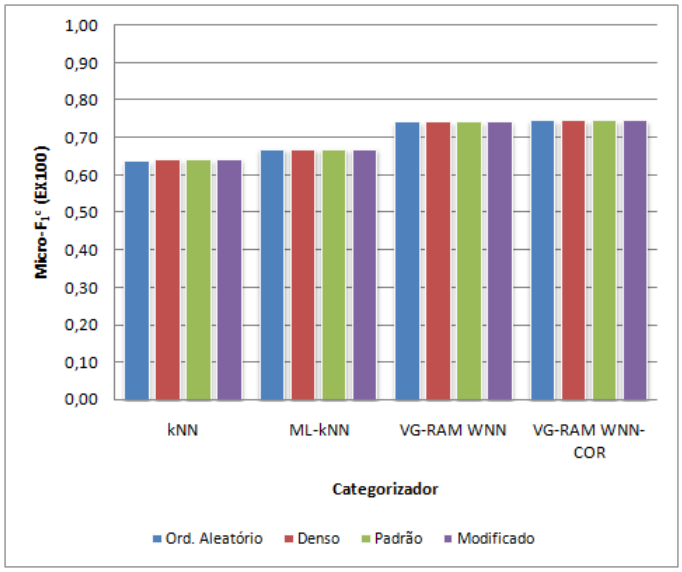
Tabela 4.12 – A estatística *t* da comparação do desempenho com o *ranking* Ordinal Aleatório, com o desempenho com os demais tipos de *ranking* segundo $macro - F_1^c$ para as bases EX100 e AT100.

Base de dados	Categorizador	Denso	Padrão	Modificado
EX100	kNN	*-5,8072*	*-5,8072*	*-5,8072*
	ML-kNN	*-0,6801*	*-0,6801*	*-0,6801*
	VG-RAM WNN	*1,4426*	*1,4426*	*1,4426*
	VG-RAM WNN-COR	*0,5048*	*0,5048*	*0,5048*
AT100	kNN	*-5,7973*	*-5,7973*	*-5,7973*
	ML-kNN	*-0,9352*	*-0,9352*	*-0,9352*
	VG-RAM WNN	*-3,5935*	*-3,5935*	*-3,5935*
	VG-RAM WNN-COR	*-0,6738*	*-0,6738*	*-0,6738*

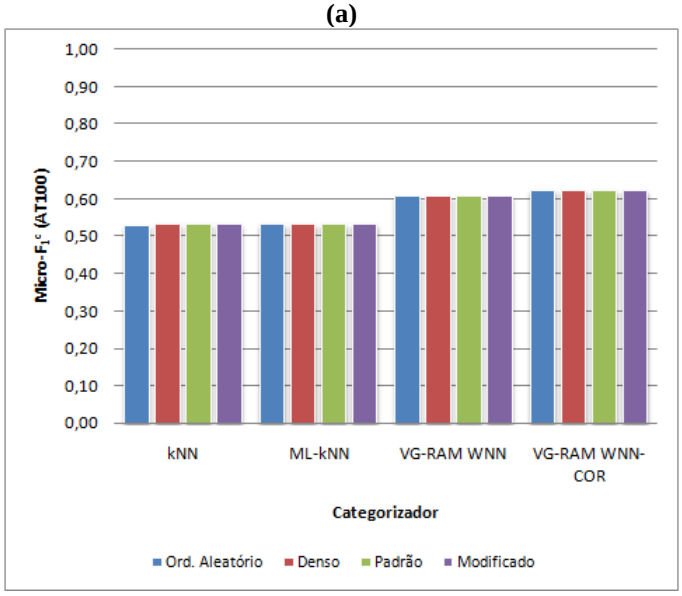
Conforme a Tabela 4.12 mostra, o desempenho do categorizador *VG-RAM WNN* segundo a métrica $macro - F_1^c$ é afetado pelos tipos de *ranking* para a base de dados AT100. O valor de $macro - F_1^c$ com o *ranking* Ordinal Aleatório é significativamente menor que com o *ranking* Denso, Padrão e Modificado.

Os resultados obtidos com os categorizadores *k NN*, *ML-k NN*, *VG-RAM WNN* e *VG-RAM WNN-COR* para a métrica $macro - F_1^c$ mostram que o *ranking* mais apropriado é o Denso, o Padrão, ou o Modificado. O *ranking* Ordinal Aleatório não é o mais apropriado, devido aos resultados obtidos para a métrica $macro - recall^c$.

A Figura 4.13 mostra de forma gráfica o impacto de cada tipo de *ranking* na métrica $micro - F_1^c$ para a base EX100 (Figura 4.13(a)) e AT100 (Figura 4.13(b)). Esta figura segue o mesmo formato da Figura 4.1.



(a)



(b)

Figura 4.13 – Resultado da métrica $micro - F_1^c$ para a base EX100, (a), e AT100, (b). Quanto maior, melhor.

Conforme as barras do gráfico da Figura 4.13(a) mostram, o valor de $micro - F_1^c$ do categorizador $k\,NN$ com a base EX100 é impactado pelo tipo de *ranking* empregado. O valor de $micro - F_1^c$ com o *ranking* Ordinal Aleatório é significativamente menor que com o *ranking* Denso, Padrão e Modificado. Entretanto, para a base de dados AT100 (Figura 4.13(b)), o desempenho deste categorizador segundo a métrica $micro - F_1^c$ não é impactado pelos tipos de *ranking* empregado.

A análise do desempenho do categorizador $ML\text{-}k\,NN$ segundo a métrica $micro - F_1^c$ mostra que o desempenho deste categorizador também não é afetado pelo tipo de *ranking* para

a base de dados EX100 (Figura 4.13(a)). O mesmo ocorre com os categorizadores *VG-RAM WNN* e *VG-RAM WNN-COR* com esta base, e com todos os categorizadores com a base de dados AT100, conforme mostra a Tabela 4.13, detalhada a seguir.

Como na Tabela 4.1, a Tabela 4.13 apresenta a estatística t associada à comparação do desempenho dos categorizadores com o *ranking* Ordinal Aleatório, com o desempenho com os demais tipos de *ranking* em estudo segundo a métrica $micro - F_1^c$ para cada uma das bases de dados.

Tabela 4.13 – A estatística *t* da comparação do desempenho com o *ranking* Ordinal Aleatório, com o desempenho com os demais tipos de *ranking* segundo $micro - F_1^c$ para as bases EX100 e AT100.

Base de dados	Categorizador	Denso	Padrão	Modificado
EX100	kNN	*-3,1258*	*-3,1258*	*-3,1258*
	ML-kNN	*-0,3869*	*-0,3869*	*-0,3869*
	VG-RAM WNN	*1,8865*	*1,8865*	*1,8865*
	VG-RAM WNN-COR	*2,0631*	*2,0631*	*2,0631*
AT100	kNN	*-2,1015*	*-2,1015*	*-2,1015*
	ML-kNN	*-1,1798*	*-1,1798*	*-1,1798*
	VG-RAM WNN	*0,9536*	*0,9536*	*0,9536*
	VG-RAM WNN-COR	*1,4592*	*1,4592*	*1,4592*

Conforme a Tabela 4.13 mostra, o desempenho de todos os categorizadores segundo a métrica $micro - F_1^c$ não é impactado pelos tipos de *ranking* Denso, Padrão e Modificado para as bases de dados EX100 e AT100, com exceção do categorizador $k\,NN$ para a base de dados EX100.

Os resultados obtidos com os categorizadores $k\,NN$, $ML\text{-}k\,NN$, *VG-RAM WNN* e *VG-RAM WNN-COR* para a métrica $micro - F_1^c$ mostram que o *ranking* mais apropriado é o Denso, o Padrão, ou o Modificado. O *ranking* Ordinal Aleatório não é o mais apropriado, conforme discutido anteriormente para a métrica $macro - F_1^c$.

4.2.6 Precisão (*precision*) orientada a documento

A métrica **precisão *(precision)* orientada a documento (** $precision_j^d$ **)** avalia a fração de categorias preditas que são pertinentes ao documento de teste d_j. A formulação é mostrada na Equação (4.28).

$$precision_j^d = \frac{\left|\hat{C}_j^{|C_j|} \cap C_j\right|}{|\hat{C}_j^{|C_j|}|} \tag{4.28}$$

O valor de $precision_j^d$ também pode ser computado usando a tabela de contingência do documento d_j (Tabela 4.14), conforme Equação (4.29).

$$precision_j^d = \frac{TP_j}{TP_j + FP_j} \tag{4.29}$$

onde FP_j (falsos positivos para d_j) é o número de categorias que foram incorretamente preditas para d_j, TN_j (verdadeiros negativos para d_j) é o número de categorias que foram corretamente não preditas para d_j; TP_j (verdadeiros positivos para d_j) é o número de categorias que foram corretamente preditas para d_j; e FN_j (falsos negativos para d_j) é o número de categorias que foram incorretamente não preditas para d_j.

Tabela 4.14 – Tabela de contingência do documento d_j.

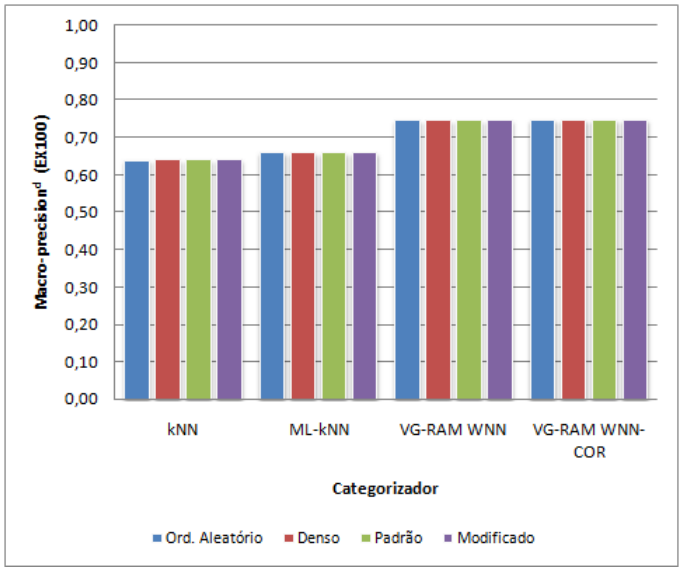
Documento d_j		**Julgamentos do especialista**	
		SIM	NÃO
Julgamentos do categorizador	SIM	TP_j	FP_j
	NÃO	FN_j	TN_j

O desempenho global de *precision* orientada a documento é calculado pelos métodos $macro - precision^d$ e $micro - precision^d$, Equação (4.30) e Equação (4.31), respectivamente. Quanto maior o valor de $macro - precision^d$ e $micro - precision^d$, melhor o desempenho do categorizador. O desempenho é perfeito quando $macro - precision^d = 1$ e $micro - precision^d = 1$.

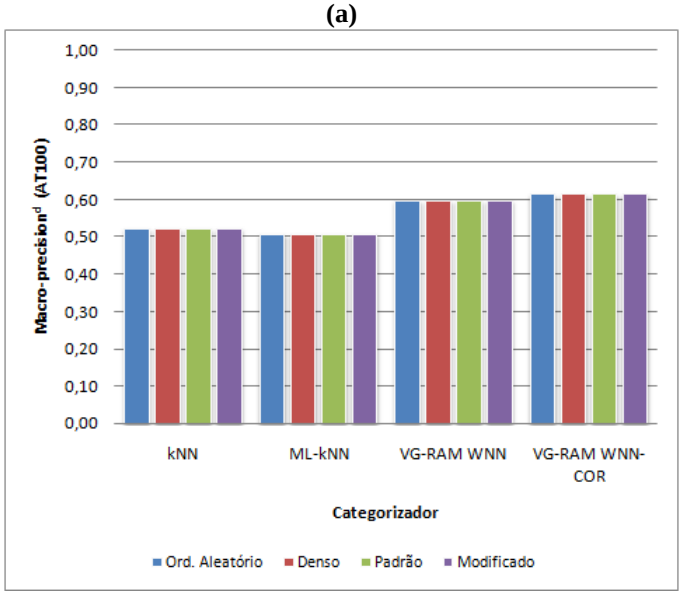
$$macro - precision^d = \frac{\sum_{j=1}^{|Te|} precision_j^d}{|Te|} \tag{4.30}$$

$$micro - precision^d = \frac{\sum_{j=1}^{|Te|} TP_j}{\sum_{j=1}^{|Te|} (TP_j + FP_j)} \tag{4.31}$$

A Figura 4.14 mostra de forma gráfica o impacto de cada tipo de *ranking* na métrica $macro-precision^d$ para a base EX100 (Figura 4.14(a)) e AT100 (Figura 4.14(b)). Esta figura segue o mesmo formato da Figura 4.1.



(a)



(b)

Figura 4.14 – Resultado da métrica $macro-precision^d$ para a base EX100, (a), e AT100, (b). Quanto maior, melhor.

Conforme as barras do gráfico da Figura 4.14(a) mostram, o valor de $macro-precision^d$ do categorizador $k\,NN$ com a base EX100 não é impactado pelo tipo de *ranking* empregado. O mesmo ocorre com os categorizadores $ML\text{-}k\,NN$, *VG-RAM WNN* e *VG-RAM WNN-COR* com esta base, e com todos os categorizadores com a base de dados AT100 (Figura 4.14(b)).

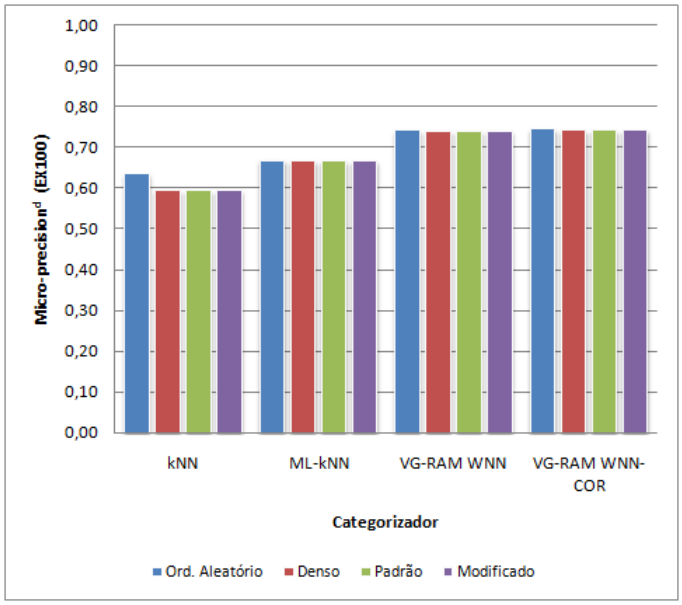
Como na Tabela 4.1, a Tabela 4.15 apresenta a estatística t associada à comparação do desempenho dos categorizadores com o *ranking* Ordinal Aleatório, com o desempenho com os demais tipos de *ranking* em estudo segundo a métrica $macro-precision^d$ para cada uma das bases de dados.

Tabela 4.15 – A estatística *t* da comparação do desempenho com o *ranking* Ordinal Aleatório, com o desempenho com os demais tipos de *ranking* segundo $macro-precision^d$ para as bases EX100 e AT100.

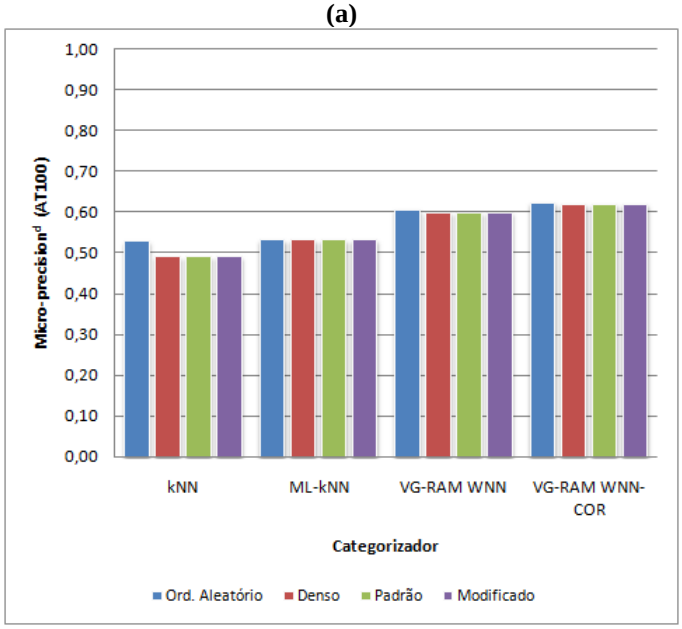
Base de dados	Categorizador	Denso	Padrão	Modificado
EX100	kNN	*-0,3577*	*-0,3577*	*-0,3577*
	ML-kNN	*1,9845*	*1,9845*	*1,9845*
	VG-RAM WNN	*1,5612*	*1,5612*	*1,5612*
	VG-RAM WNN-COR	*0,7982*	*0,7982*	*0,7982*
AT100	kNN	*-0,0711*	*-0,0711*	*-0,0711*
	ML-kNN	*0,2189*	*0,2189*	*0,2189*
	VG-RAM WNN	*0,4723*	*0,4723*	*0,4723*
	VG-RAM WNN-COR	*0,5481*	*0,5481*	*0,5481*

Conforme a Tabela 4.15, o desempenho de todos os categorizadores segundo a métrica $macro-precision^d$ não é impactado pelos tipos de *ranking* Denso, Padrão e Modificado para as bases de dados EX100 e AT100. Então, o *ranking* mais apropriado para a métrica $macro-precision^d$ é o *ranking* o Ordinal Aleatório, o Denso, o Padrão ou o Modificado.

A Figura 4.15 mostra de forma gráfica o impacto de cada tipo de *ranking* na métrica $micro-precision^d$ para a base EX100 (Figura 4.15(a)) e AT100 (Figura 4.15(b)). Esta figura segue o mesmo formato da Figura 4.1.



(a)



(b)

Figura 4.15 – Resultado da métrica $micro-precision^d$ para a base EX100, (a), e AT100, (b). Quanto maior, melhor.

Conforme as barras do gráfico da Figura 4.15(a) mostram, o valor de $micro-precision^d$ do categorizador $k\,NN$ com a base EX100 é impactado pelo tipo de *ranking* empregado. O valor de $micro-precision^d$ com o *ranking* Ordinal Aleatório é significativamente maior que com o *ranking* Denso, Padrão e Modificado (teste t pareado bicaudal com nível de significância 5%). O mesmo ocorre com os categorizadores *VG-RAM WNN* e *VG-RAM WNN-COR* com esta base, e com os categorizadores $k\,NN$, *VG-RAM WNN* e *VG-RAM WNN-COR* com a base de dados AT100 (Figura 4.15(b)).

A análise do desempenho do categorizador *ML-k NN* segundo a métrica $micro-precision^d$ mostra que o desempenho deste categorizador é afetado pelo tipo de *ranking* para a base de dados EX100 (Figura 4.15(a)). O valor de $micro-precision^d$ com o *ranking* Ordinal Aleatório é significativamente maior que com o *ranking* Denso, Padrão e Modificado, conforme mostra a Tabela 4.16, detalhada a seguir.

Como na Tabela 4.1, a Tabela 4.16 apresenta a estatística t associada à comparação do desempenho dos categorizadores com o *ranking* Ordinal Aleatório, com o desempenho com os demais tipos de *ranking* em estudo segundo a métrica $micro-precision^d$ para cada uma das bases de dados.

Tabela 4.16 – A estatística *t* da comparação do desempenho com o *ranking* Ordinal Aleatório, com o desempenho com os demais tipos de *ranking* segundo $micro-precision^d$ para as bases EX100 e AT100.

Base de dados	Categorizador	Denso	Padrão	Modificado
EX100	kNN	*20,8092*	*20,8092*	*20,8092*
	ML-kNN	*3,6645*	*3,6645*	*3,6645*
	VG-RAM WNN	*5,9509*	*5,9509*	*5,9509*
	VG-RAM WNN-COR	*11,6648*	*11,6648*	*11,6648*
AT100	kNN	*16,0190*	*16,0190*	*16,0190*
	ML-kNN	*0,9505*	*0,9505*	*0,9505*
	VG-RAM WNN	*10,2075*	*10,2075*	*10,2075*
	VG-RAM WNN-COR	*5,9143*	*5,9143*	*5,9143*

Como a Tabela 4.16 mostra, o desempenho do categorizador *ML-k NN* não é afetado pelo tipo de *ranking* para a base de dados AT100. Ou seja, o valor de $micro-precision^d$ deste categorizador com o *ranking* Ordinal Aleatório não é significativamente diferente que com os *rankings* Denso, Padrão e Modificado.

Os resultados obtidos com os categorizadores *k NN*, *ML-k NN*, *VG-RAM WNN* e *VG-RAM WNN-COR* para a métrica $micro-precision^d$ mostram que o *ranking* mais apropriado é o Denso, o Padrão, ou o Modificado. O *ranking* Ordinal Aleatório não é o mais apropriado, pois favoreceu os categorizadores que geraram *rankings* com empates entre as categorias de interesse no conjunto $\hat{C}_j^{|C_j|}$, com exceção do categorizador *ML-k NN* para a base de dados AT100.

4.2.7 Revocação (*recall*) orientada a documento

A métrica **revocação (*recall*) orientada a documento (** $recall_j^d$ **)** avalia a fração de categorias pertinentes que são preditas para o documento de teste d_j. A formulação é apresentada na Equação (4.32).

$$recall_j^d = \frac{\left|\hat{C}_j^{|C_j|} \cap C_j\right|}{|C_j|} \tag{4.32}$$

O valor de $recall_j^d$ pode também ser obtido em termos da tabela de contingência do documento d_j (Tabela 4.14) conforme a Equação (4.33).

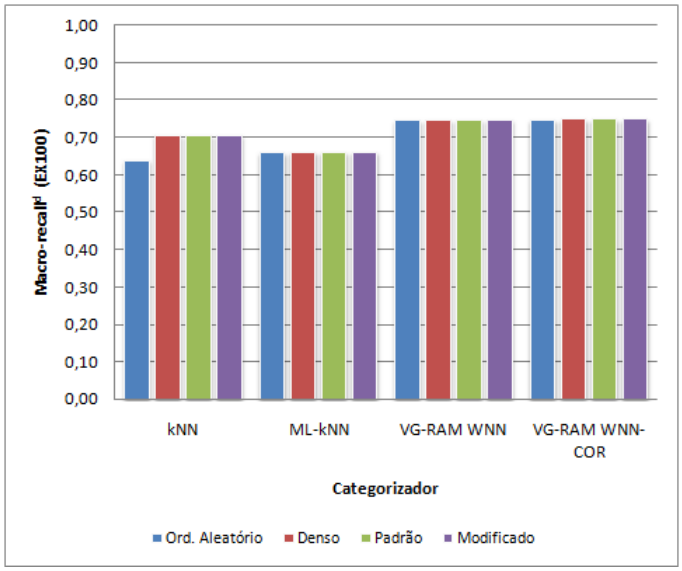
$$recall_j^d = \frac{TP_j}{TP_j + FN_j} \tag{4.33}$$

O desempenho global de *recall* orientado a documento é calculado pelos métodos $macro - recall^d$ e $micro - recall^d$, Equação (4.34) e Equação (4.35), respectivamente. Quanto maior o valor de $macro - recall^d$ e $micro - recall^d$, melhor o desempenho do categorizador. O desempenho é perfeito quando $macro - recall^d = 1$ e $micro - recall^d = 1$.

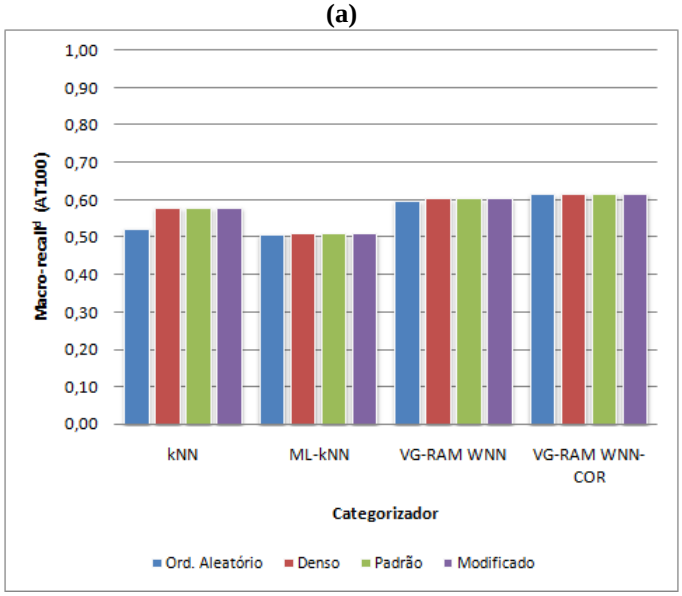
$$macro - recall^d = \frac{\sum_{j=1}^{|Te|} recall_j^d}{|Te|} \tag{4.34}$$

$$micro - recall^d = \frac{\sum_{j=1}^{|Te|} TP_j}{\sum_{j=1}^{|Te|} \left(TP_j + FN_j\right)} \tag{4.35}$$

A Figura 4.16 mostra de forma gráfica o impacto de cada tipo de *ranking* na métrica $macro - recall^d$ para a base EX100 (Figura 4.16(a)) e AT100 (Figura 4.16(b)). Esta figura segue o mesmo formato da Figura 4.1.



(a)



(b)

Figura 4.16 – Resultado da métrica $macro-recall^d$ para a base EX100, (a), e AT100, (b). Quanto maior, melhor.

Conforme as barras do gráfico da Figura 4.16(a) mostram, o valor de $macro-recall^d$ do categorizador $k\,NN$ com a base EX100 é impactado pelo tipo de *ranking* empregado. O valor de $macro-recall^d$ com o *ranking* Ordinal Aleatório é significativamente menor que com o *ranking* Denso, Padrão e Modificado (teste t pareado bicaudal com nível de significância 5%). O mesmo ocorre com o categorizador *VG-RAM WNN-COR* com esta base, e os categorizadores $k\,NN$, $ML\text{-}k\,NN$ e *VG-RAM WNN* com a base de dados AT100 (Figura 4.16(b)).

A análise do desempenho do categorizador $ML\text{-}k\,NN$ segundo a métrica $macro-recall^d$ mostra que o desempenho deste categorizador também é afetado pelo tipo de

ranking para a base de dados EX100 (Figura 4.16(a)). O valor de $macro-recall^d$ deste categorizador com o *ranking* Ordinal Aleatório significativamente menor que com os *rankings* Denso, Padrão e Modificado, conforme mostra a Tabela 4.17, detalhada a seguir.

Como na Tabela 4.1, a Tabela 4.17 apresenta a estatística t associada à comparação do desempenho dos categorizadores com o *ranking* Ordinal Aleatório, com o desempenho com os demais tipos de *ranking* em estudo segundo a métrica $macro-recall^d$ para cada uma das bases de dados.

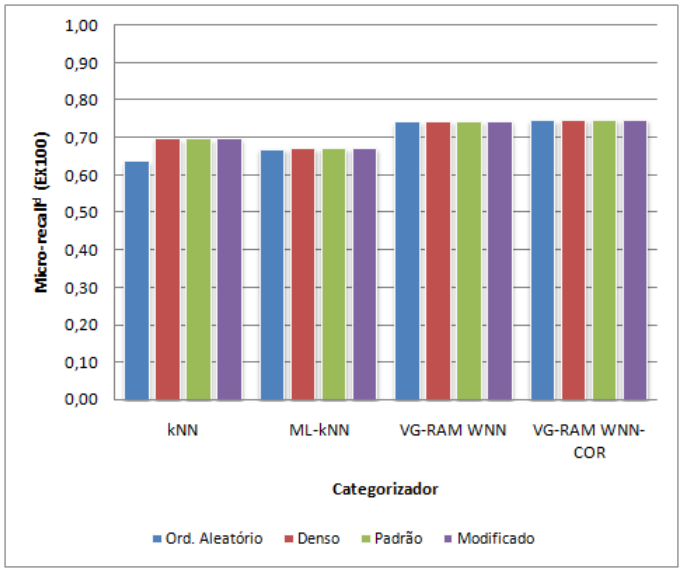
Tabela 4.17 – A estatística *t* da comparação do desempenho com o *ranking* Ordinal Aleatório, com o desempenho com os demais tipos de *ranking* segundo $macro-recall^d$ para as bases EX100 e AT100.

Base de dados	Categorizador	Denso	Padrão	Modificado
EX100	kNN	*-31,5611*	*-31,5611*	*-31,5611*
	ML-kNN	*-5,6691*	*-5,6691*	*-5,6691*
	VG-RAM WNN	*-3,0362*	*-3,0362*	*-3,0362*
	VG-RAM WNN-COR	*-4,8384*	*-4,8384*	*-4,8384*
AT100	kNN	*-21,9213*	*-21,9213*	*-21,9213*
	ML-kNN	*-2,9890*	*-2,9890*	*-2,9890*
	VG-RAM WNN	*-9,5149*	*-9,5149*	*-9,5149*
	VG-RAM WNN-COR	*-4,4775*	*-4,4775*	*-4,4775*

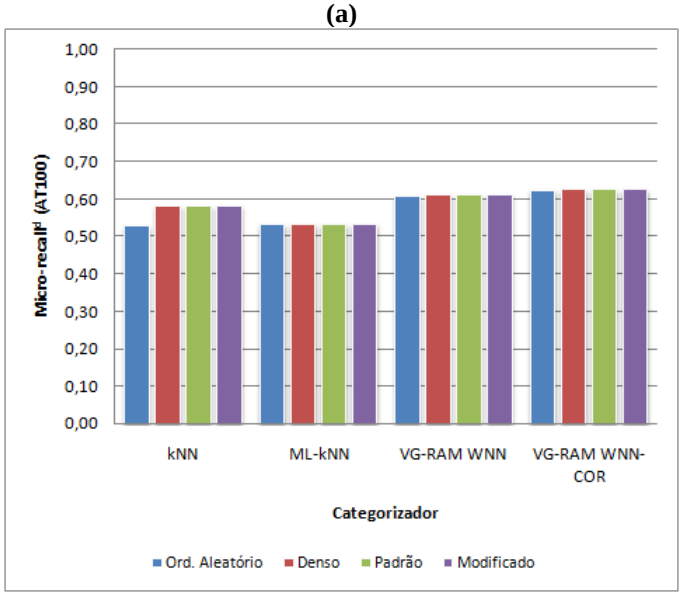
Como a Figura 4.16 mostra, o impacto do tipo de *ranking* no desempenho dos categorizadores *VG-RAM WNN* e *VG-RAM WNN-COR* é similar àquele observado no categorizador *ML-k NN*. Conforme a Tabela 4.17 mostra, o desempenho de todos os categorizadores segundo a métrica $macro-recall^d$ é impactado pelos tipos de *ranking* Denso, Padrão e Modificado para as bases de dados EX100 e AT100. O valor de $macro-recall^d$ destes categorizadores com o *ranking* Ordinal Aleatório é significativamente menor que com os *rankings* Denso e Padrão e Modificado.

Os resultados obtidos com os categorizadores *k NN*, *ML-k NN*, *VG-RAM WNN* e *VG-RAM WNN-COR* para a métrica $macro-recall^d$ mostram que o *ranking* mais apropriado é o Denso, o Padrão, ou o Modificado. O *ranking* Ordinal Aleatório não é o mais apropriado, conforme discutido para a métrica $macro-recall^c$.

A Figura 4.17 mostra de forma gráfica o impacto de cada tipo de *ranking* na métrica $micro-recall^d$ para a base EX100 (Figura 4.17(a)) e AT100 (Figura 4.17(b)). Esta figura segue o mesmo formato da Figura 4.1.



(a)



(b)

Figura 4.17 – Resultado da métrica $micro-recall^d$ para a base EX100, (a), e AT100, (b). Quanto maior, melhor.

O desempenho dos categorizadores *k NN*, *ML-k NN*, *VG-RAM WNN* e *VG-RAM WNN-COR* segundo a métrica $micro-recall^d$ é impactado pelos tipos de *ranking* em estudo para as duas bases de dados, de forma similar àquele observado na métrica $macro-recall^d$, conforme mostra a Tabela 4.18.

A Tabela 4.18 mostra a estatística t associada à comparação do desempenho dos categorizadores com o *ranking* Ordinal Aleatório, com o desempenho com os demais tipos de *ranking* em estudo segundo a métrica $micro-recall^d$ para cada uma das bases de dados.

Tabela 4.18 – A estatística *t* da comparação do desempenho com o *ranking* Ordinal Aleatório, com o desempenho com os demais tipos de *ranking* segundo $micro-recall^d$ para as bases EX100 e AT100.

Base de dados	Categorizador	Denso	Padrão	Modificado
EX100	kNN	*-32,2633*	*-32,2633*	*-32,2633*
	ML-kNN	*-6,3973*	*-6,3973*	*-6,3973*
	VG-RAM WNN	*-3,4920*	*-3,4920*	*-3,4920*
	VG-RAM WNN-COR	*-5,3933*	*-5,3933*	*-5,3933*
AT100	kNN	*-28,0168*	*-28,0168*	*-28,0168*
	ML-kNN	*-2,8868*	*-2,8868*	*-2,8868*
	VG-RAM WNN	*-9,7551*	*-9,7551*	*-9,7551*
	VG-RAM WNN-COR	*-5,8698*	*-5,8698*	*-5,8698*

De acordo com a Tabela 4.18, o desempenho de todos os categorizadores segundo a métrica $micro-recall^d$ é impactado pelos tipos de *ranking* Denso, Padrão e Modificado para as bases de dados EX100 e AT100. O valor de $micro-recall^d$ destes categorizadores com o *ranking* Ordinal Aleatório é significativamente menor que com os *rankings* Denso, Padrão e Modificado.

Os resultados obtidos com os categorizadores *k NN*, *ML-k NN*, *VG-RAM WNN* e *VG-RAM WNN-COR* para a métrica $micro-recall^d$ mostram que o *ranking* mais apropriado é o Denso, o Padrão, ou o Modificado. O *ranking* Ordinal Aleatório não é o mais apropriado, conforme discutido para a métrica $macro-recall^c$.

4.2.8 F_β orientada a documento

A métrica F_β **orientada a documento (** ${F_\beta}_j^d$ **)** avalia a média harmônica ponderada de $precision_j^d$ e $recall_j^d$. A formulação original de Rijsbergen [Rijsbergen79] é mostrada na Equação (4.36).

$$ {F_\beta}_j^d = \frac{(\beta^2+1) * precision_j^d * recall_j^d}{\beta^2 * precision_j^d + recall_j^d} \tag{4.36} $$

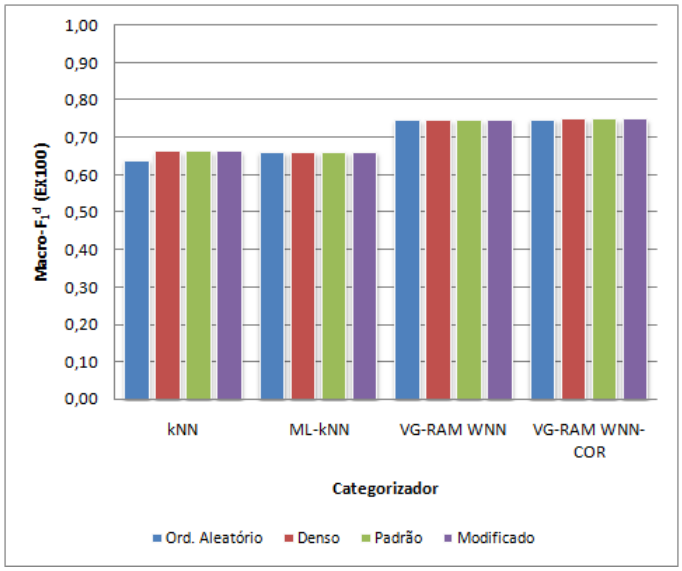
Como na métrica F_β orientada à categoria, $\beta = 1$ é utilizado, atribuindo importância igual para $precision_j^d$ e $recall_j^d$. O desempenho global de F_1^d é computado pelos métodos $macro-F_1^d$ (Equação (4.37)) e $micro-F_1^d$ (Equação (4.38)). Quanto maior o valor de

$macro - F_1^{\,d}$ e $micro - F_1^{\,d}$, melhor o desempenho do categorizador. O desempenho é perfeito quando $macro - F_1^{\,d} = 1$ e $micro - F_1^{\,d} = 1$.

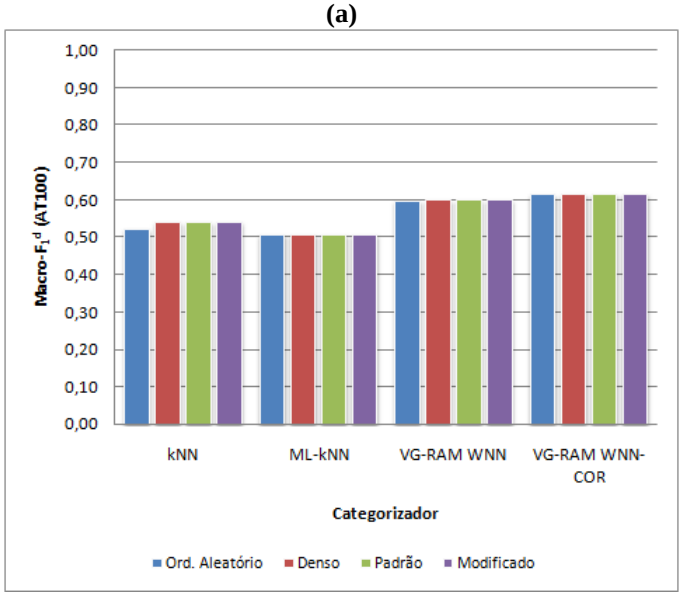
$$macro - F_1^{\,d} = \frac{1}{|Te|}\sum_{j=1}^{|Te|} F_{1\,j}^{\,d} \tag{4.37}$$

$$micro - F_1^{\,d} = \frac{2 * micro - precision_j^d * micro - recall_j^d}{micro - precision_j^d + micro - recall_j^d} \tag{4.38}$$

A Figura 4.18 mostra de forma gráfica o impacto de cada tipo de *ranking* na métrica $macro - F_1^{\,d}$ para a base EX100 (Figura 4.18(a)) e AT100 (Figura 4.18(b)). Esta figura segue o mesmo formato da Figura 4.1.



(a)



(b)

Figura 4.18 – Resultado da métrica $macro - F_1^d$ para a base EX100, (a), e AT100, (b). Quanto maior, melhor.

Conforme as barras do gráfico da Figura 4.18(a) mostram, o valor de $macro - F_1^d$ do categorizador $k\,NN$ com a base EX100 é impactado pelo tipo de *ranking* empregado. O valor de $macro - F_1^d$ com o *ranking* Ordinal Aleatório é significativamente menor que com o *ranking* Denso, Padrão e Modificado (teste t pareado bicaudal com nível de significância 5%). O mesmo ocorre com os categorizadores $k\,NN$ e *VG-RAM WNN* com a base de dados AT100 (Figura 4.18(b)).

A análise do desempenho do categorizador *ML-*$k\,NN$ segundo a métrica $macro - F_1^d$ mostra que o desempenho deste categorizador não é afetado pelo tipo de *ranking* para a base

de dados EX100 (Figura 4.12(a)). O mesmo ocorre com os categorizadores *VG-RAM WNN* e *VG-RAM WNN-COR* com esta base, conforme mostra a Tabela 4.19, detalhada a seguir.

Como na Tabela 4.1, a Tabela 4.19 apresenta a estatística t associada à comparação do desempenho dos categorizadores com o *ranking* Ordinal Aleatório, com o desempenho com os demais tipos de *ranking* em estudo segundo a métrica $macro - F_1^d$ para cada uma das bases de dados.

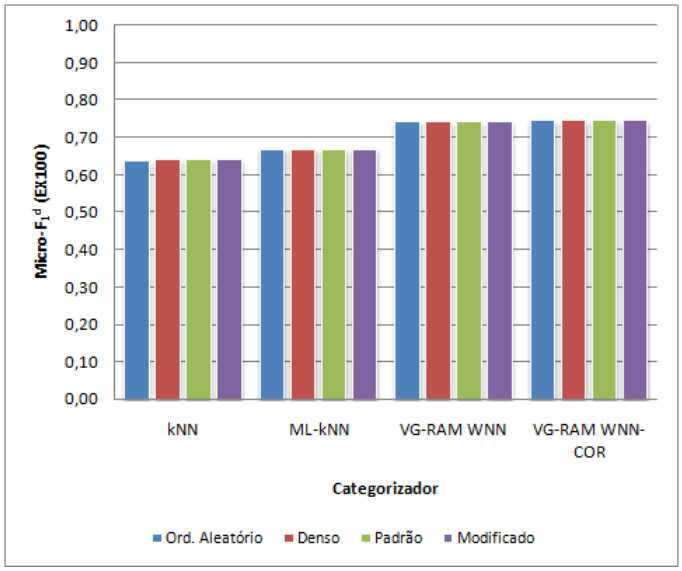
Tabela 4.19 – A estatística *t* da comparação do desempenho com o *ranking* Ordinal Aleatório, com o desempenho com os demais tipos de *ranking* segundo $macro - F_1^d$ para as bases EX100 e AT100.

Base de dados	Categorizador	Denso	Padrão	Modificado
EX100	kNN	*-12,4441*	*-12,4441*	*-12,4441*
	ML-kNN	*-0,8737*	*-0,8737*	*-0,8737*
	VG-RAM WNN	*-1,0394*	*-1,0394*	*-1,0394*
	VG-RAM WNN-COR	*-2,0726*	*-2,0726*	*-2,0726*
AT100	kNN	*-9,1851*	*-9,1851*	*-9,1851*
	ML-kNN	*-1,3261*	*-1,3261*	*-1,3261*
	VG-RAM WNN	*-5,0338*	*-5,0338*	*-5,0338*
	VG-RAM WNN-COR	*-2,7186*	*-2,7186*	*-2,7186*

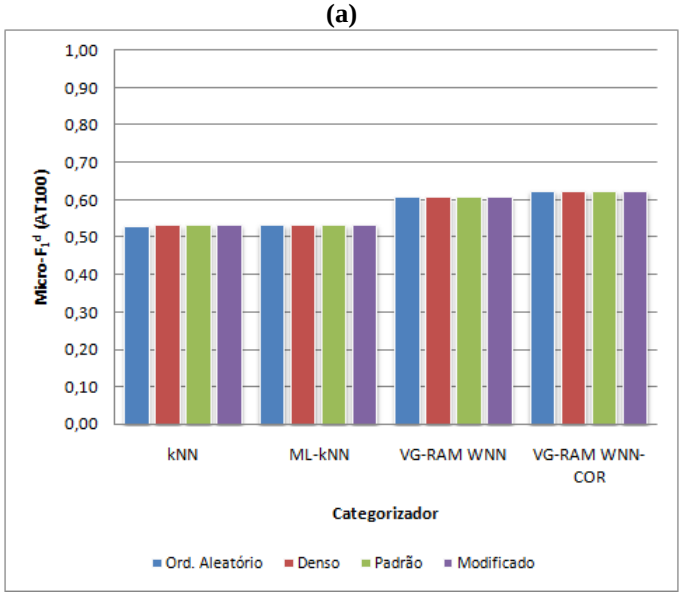
Como a Figura 4.12 mostra, o impacto do tipo de *ranking* no desempenho dos categorizadores *ML-k NN* e *VG-RAM WNN-COR* é similar àquele observado no categorizador *VG-RAM WNN*, muito embora não seja idêntico. De acordo com a Tabela 4.19, o desempenho do categorizador *ML-k NN* segundo a métrica $macro - F_1^d$ não é afetado pelo tipo de *ranking* para a base de dados AT100. Entretanto, o mesmo não ocorre com o categorizador *VG-RAM WNN-COR* para esta base. O valor de $macro - F_1^d$ deste categorizador com o *ranking* Ordinal Aleatório é significativamente menor que com os *rankings* Denso, Padrão e Modificado.

Os resultados obtidos com os categorizadores *k NN*, *ML-k NN*, *VG-RAM WNN* e *VG-RAM WNN-COR* para a métrica $macro - F_1^d$ mostram que o *ranking* mais apropriado é o Denso, o Padrão, ou o Modificado. O *ranking* Ordinal Aleatório não é o mais apropriado, conforme discutido para a métrica $macro - F_1^c$.

A Figura 4.19 mostra de forma gráfica o impacto de cada tipo de *ranking* na métrica $micro - F_1^d$ para a base EX100 (Figura 4.19(a)) e AT100 (Figura 4.19(b)). Esta figura segue o mesmo formato da Figura 4.1.



(a)



(b)

Figura 4.19 – Resultado da métrica $micro - F_1^d$ para a base EX100, (a), e AT100, (b). Quanto maior, melhor.

Conforme as barras do gráfico da Figura 4.19(a) mostram, o valor de $micro - F_1^d$ do categorizador $k\,NN$ com a base EX100 é impactado pelo tipo de *ranking* empregado. O valor de $micro - F_1^d$ com o *ranking* Ordinal Aleatório é significativamente menor que com o *ranking* Denso, Padrão e Modificado Entretanto, para a base de dados AT100 (Figura 4.19(b)), o valor de $micro - F_1^c$ com o *ranking* Ordinal Aleatório não é significativamente diferente que com o *ranking* Denso, Padrão e Modificado.

A análise do desempenho do categorizador *ML-k NN* segundo a métrica $micro - F_1^d$ mostra que o desempenho deste categorizador também não é afetado pelo tipo de *ranking* para

a base de dados EX100 (Figura 4.19(a)). O mesmo ocorre com os categorizadores *VG-RAM WNN* e *VG-RAM WNN-COR* com esta base, e com os categorizadores *ML-k NN*, *VG-RAM WNN* e *VG-RAM WNN-COR* com a base de dados AT100, conforme mostra a Tabela 4.20, detalhada a seguir.

Como na Tabela 4.1, a Tabela 4.20 apresenta a estatística *t* associada à comparação do desempenho dos categorizadores com o *ranking* Ordinal Aleatório, com o desempenho com os demais tipos de *ranking* em estudo segundo a métrica $micro - F_1^d$ para cada uma das bases de dados.

Tabela 4.20 – A estatística *t* da comparação do desempenho com o *ranking* Ordinal Aleatório, com o desempenho com os demais tipos de *ranking* segundo $micro - F_1^d$ para as bases EX100 e AT100.

Base de dados	Categorizador	Denso	Padrão	Modificado
EX100	kNN	*-3,1258*	*-3,1258*	*-3,1258*
	ML-kNN	*-0,3869*	*-0,3869*	*-0,3869*
	VG-RAM WNN	*1,8865*	*1,8865*	*1,8865*
	VG-RAM WNN-COR	*2,0631*	*2,0631*	*2,0631*
AT100	kNN	*-2,1015*	*-2,1015*	*-2,1015*
	ML-kNN	*-1,1798*	*-1,1798*	*-1,1798*
	VG-RAM WNN	*0,9536*	*0,9536*	*0,9536*
	VG-RAM WNN-COR	*1,4592*	*1,4592*	*1,4592*

Conforme a Tabela 4.20 mostra, o desempenho de todos os categorizadores segundo a métrica $micro - F_1^d$ não é impactado pelos tipos de *ranking* Denso, Padrão e Modificado para as bases de dados EX100 e AT100, com exceção do categorizador *k NN* para a base de dados EX100.

Os resultados obtidos com os categorizadores *k NN*, *ML-k NN*, *VG-RAM WNN* e *VG-RAM WNN-COR* para a métrica $micro - F_1^d$ mostram que o *ranking* mais apropriado é o Denso, o Padrão, ou o Modificado. O *ranking* Ordinal Aleatório não é o mais apropriado, conforme discutido para a métrica $macro - F_1^c$.

Note que o desempenho dos categorizadores pelo método *microaveraging* dá resultado igual, independente de ser definida orientada à categoria ou a documento. A expansão das formulações de $micro - precision^c$ e $micro - precision^d$ é mostrada na Equação (4.39) e Equação (4.40), respectivamente.

$$micro-precision^{c} = \frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|}\left(TP_i + FP_i\right)} = \frac{\sum_{i=1}^{|C|}\sum_{j=1}^{|Te|} TP_{ij}}{\sum_{i=1}^{|C|}\left(\sum_{j=1}^{|Te|} TP_{ij} + \sum_{j=1}^{|Te|} FP_{ij}\right)} \quad (4.39)$$

$$micro-precision^{d} = \frac{\sum_{j=1}^{|Te|} TP_j}{\sum_{j=1}^{|Te|}\left(TP_j + FP_j\right)} = \frac{\sum_{j=1}^{|Te|}\sum_{i=1}^{|C|} TP_{ij}}{\sum_{j=1}^{|Te|}\left(\sum_{i=1}^{|C|} TP_{ij} + \sum_{i=1}^{|C|} FP_{ij}\right)} \quad (4.40)$$

Observa-se pela Equação (4.39) e Equação (4.40), $micro-precision^{c}$ é igual a $micro-precision^{d}$. Analogamente, $micro-recall^{c}$ e $micro-F_1{}^{c}$ são iguais a $micro-recall^{d}$ e $micro-F_1{}^{d}$, respectivamente.

5 DISCUSSÃO

No Capítulo 4 examinamos de forma experimental o impacto dos tipos de *rankings* Ordinal Aleatório, Denso, Padrão e Modificado no desempenho dos categorizadores multi-rótulo de texto *k NN*, *ML-k NN*, *VG-RAM WNN* e *VG-RAM WNN-COR*, segundo as métricas *one-error*, *coverage*, *ranking loss*, *average precision*, *R-precision*, *Hamming loss*, *exact match*, *precision*, *recall* e F_1, com duas bases de dados (EX100 e AT100) contendo documentos texto de descrições de atividades econômicas de empresas brasileiras. As definições das métricas *one-error* e *average precision* foram alteradas de modo a comportar o tratamento de empates pelos *rankings* Denso, Padrão e Modificado. As definições originais das demais métricas não foram alteradas, pois permitem trabalhar com esses *rankings*.

Os resultados obtidos no Capítulo 4 mostram que, dependendo do tipo de *ranking* empregado, o desempenho de um categorizador segundo uma determinada métrica é significativamente diferente, e isso foi observado para a maioria das métricas analisadas neste trabalho.

Para apresentar uma visão clara do impacto dos tipos de *ranking* no desempenho dos categorizadores examinados, uma ordem parcial é definida para cada métrica, onde $O \prec D$ ou $O \succ D$ significa que o desempenho do categorizador com o *ranking* Ordinal Aleatório é significativamente diferente (teste t pareado bicaudal com nível de significância 5%) que com o *ranking* Denso. Se o desempenho não é significativamente diferente, a ordem parcial $O \equiv D$ é utilizada. A mesma representação da ordem parcial também é empregada para os *rankings* Padrão (P) e Modificado (M).

A Tabela 5.1 mostra um sumário dos resultados obtidos no Capítulo 4 para a base de dados EX100 utilizando a representação de ordem parcial. Como os resultados dos *rankings* Denso, Padrão e Modificados são iguais para todas as métricas de avaliação consideradas, com exceção da métrica *coverage*, é utilizado uma representação única da ordem parcial nesta tabela, por exemplo, $O \prec D \mid P \mid M$, significando que $O \prec D$, $O \prec P$ e $O \prec M$.

Tabela 5.1 – Sumário dos resultados do teste *t* para a base EX100.

Métricas	***k NN***	***ML-k NN***	***VG-RAM WNN***	***VG-RAM WNN-COR***
*one-error**	$O \prec D \mid P \mid M$	$O \prec D \mid P \mid M$	$O \prec D \mid P \mid M$	$O \equiv D \mid P \mid M$
coverage	$O \succ D \mid P$	$O \succ D \mid P$	$O \succ D \mid P$	$O \succ D \mid P$
	$O \prec M$	$O \prec M$	$O \prec M$	$O \prec M$
ranking loss	$O \equiv D \mid P \mid M$	$O \equiv D \mid P \mid M$	$O \equiv D \mid P \mid M$	$O \equiv D \mid P \mid M$
*average precision**	$O \succ D \mid P \mid M$	$O \succ D \mid P \mid M$	$O \succ D \mid P \mid M$	$O \succ D \mid P \mid M$
R-precision	$O \equiv D \mid P \mid M$	$O \equiv D \mid P \mid M$	$O \equiv D \mid P \mid M$	$O \equiv D \mid P \mid M$
Hamming loss	$O \prec D \mid P \mid M$	$O \prec D \mid P \mid M$	$O \prec D \mid P \mid M$	$O \prec D \mid P \mid M$
exact match	$O \succ D \mid P \mid M$	$O \succ D \mid P \mid M$	$O \succ D \mid P \mid M$	$O \succ D \mid P \mid M$
$macro - precision^{c}$	$O \succ D \mid P \mid M$	$O \succ D \mid P \mid M$	$O \succ D \mid P \mid M$	$O \equiv D \mid P \mid M$
$micro - precision^{c}$	$O \succ D \mid P \mid M$	$O \succ D \mid P \mid M$	$O \succ D \mid P \mid M$	$O \succ D \mid P \mid M$
$macro - recall^{c}$	$O \prec D \mid P \mid M$	$O \prec D \mid P \mid M$	$O \prec D \mid P \mid M$	$O \prec D \mid P \mid M$
$micro - recall^{c}$	$O \prec D \mid P \mid M$	$O \prec D \mid P \mid M$	$O \prec D \mid P \mid M$	$O \prec D \mid P \mid M$
$macro - F_1^{c}$	$O \prec D \mid P \mid M$	$O \equiv D \mid P \mid M$	$O \equiv D \mid P \mid M$	$O \equiv D \mid P \mid M$
$micro - F_1^{c}$	$O \prec D \mid P \mid M$	$O \equiv D \mid P \mid M$	$O \equiv D \mid P \mid M$	$O \equiv D \mid P \mid M$
$macro - precision^{d}$	$O \equiv D \mid P \mid M$	$O \equiv D \mid P \mid M$	$O \equiv D \mid P \mid M$	$O \equiv D \mid P \mid M$
$micro - precision^{d}$	$O \succ D \mid P \mid M$	$O \succ D \mid P \mid M$	$O \succ D \mid P \mid M$	$O \succ D \mid P \mid M$
$macro - recall^{d}$	$O \prec D \mid P \mid M$	$O \prec D \mid P \mid M$	$O \prec D \mid P \mid M$	$O \prec D \mid P \mid M$
$micro - recall^{d}$	$O \prec D \mid P \mid M$	$O \prec D \mid P \mid M$	$O \prec D \mid P \mid M$	$O \prec D \mid P \mid M$
$macro - F_1^{d}$	$O \prec D \mid P \mid M$	$O \equiv D \mid P \mid M$	$O \equiv D \mid P \mid M$	$O \equiv D \mid P \mid M$
$micro - F_1^{d}$	$O \prec D \mid P \mid M$	$O \equiv D \mid P \mid M$	$O \equiv D \mid P \mid M$	$O \equiv D \mid P \mid M$

Como a Tabela 5.1 mostra, o tipo de *ranking* impacta na grande maioria das métricas utilizadas para avaliar o desempenho dos categorizadores para a base de dados EX100. De acordo com a Tabela 5.1, o desempenho do categorizador *k NN* é impactado pelo tipo de *ranking* empregado em 16 das 19 métricas de avaliação analisadas para a base de dados EX100. A análise do desempenho do categorizador *ML-k NN* mostra que este categorizador foi impactado em 12 métricas das 19 métricas de avaliação analisadas para a base de dados EX100. O desempenho do categorizador *VG-RAM WNN* foi também impactado em 12 métricas. Finalmente, a análise de desempenho do VG-*RAM WNN-COR* mostra que este categorizador é impactado em 10 métricas das 19 métricas de avaliação.

Como na Tabela 5.1, a Tabela 5.2 mostra um sumário dos resultados obtidos no Capítulo 4 para a base de dados AT100 utilizando a representação de ordem parcial.

Tabela 5.2 – Sumário dos resultados do teste *t* para a base AT100.

Métricas	***k NN***	***ML-k NN***	***VG-RAM WNN***	***VG-RAM WNN-COR***
*one-error**	$O \prec D \mid P \mid M$	$O \equiv D \mid P \mid M$	$O \prec D \mid P \mid M$	$O \prec D \mid P \mid M$
coverage	$O \succ D \mid P$	$O \succ D \mid P$	$O \succ D \mid P$	$O \succ D \mid P$
	$O \prec M$	$O \prec M$	$O \prec M$	$O \prec M$
ranking loss	$O \equiv D \mid P \mid M$	$O \equiv D \mid P \mid M$	$O \equiv D \mid P \mid M$	$O \equiv D \mid P \mid M$
*average precision**	$O \succ D \mid P \mid M$	$O \succ D \mid P \mid M$	$O \succ D \mid P \mid M$	$O \succ D \mid P \mid M$
R-precision	$O \equiv D \mid P \mid M$	$O \equiv D \mid P \mid M$	$O \equiv D \mid P \mid M$	$O \equiv D \mid P \mid M$
Hamming loss	$O \prec D \mid P \mid M$	$O \equiv D \mid P \mid M$	$O \prec D \mid P \mid M$	$O \prec D \mid P \mid M$
exact match	$O \succ D \mid P \mid M$	$O \equiv D \mid P \mid M$	$O \succ D \mid P \mid M$	$O \succ D \mid P \mid M$
$macro - precision^{c}$	$O \succ D \mid P \mid M$	$O \equiv D \mid P \mid M$	$O \equiv D \mid P \mid M$	$O \equiv D \mid P \mid M$
$micro - precision^{c}$	$O \succ D \mid P \mid M$	$O \equiv D \mid P \mid M$	$O \succ D \mid P \mid M$	$O \succ D \mid P \mid M$
$macro - recall^{c}$	$O \prec D \mid P \mid M$	$O \prec D \mid P \mid M$	$O \prec D \mid P \mid M$	$O \prec D \mid P \mid M$
$micro - recall^{c}$	$O \prec D \mid P \mid M$	$O \prec D \mid P \mid M$	$O \prec D \mid P \mid M$	$O \prec D \mid P \mid M$
$macro - F_1^{c}$	$O \prec D \mid P \mid M$	$O \equiv D \mid P \mid M$	$O \prec D \mid P \mid M$	$O \equiv D \mid P \mid M$
$micro - F_1^{c}$	$O \equiv D \mid P \mid M$	$O \equiv D \mid P \mid M$	$O \equiv D \mid P \mid M$	$O \equiv D \mid P \mid M$
$macro - precision^{d}$	$O \equiv D \mid P \mid M$	$O \equiv D \mid P \mid M$	$O \equiv D \mid P \mid M$	$O \equiv D \mid P \mid M$
$micro - precision^{d}$	$O \succ D \mid P \mid M$	$O \equiv D \mid P \mid M$	$O \succ D \mid P \mid M$	$O \succ D \mid P \mid M$
$macro - recall^{d}$	$O \prec D \mid P \mid M$	$O \prec D \mid P \mid M$	$O \prec D \mid P \mid M$	$O \prec D \mid P \mid M$
$micro - recall^{d}$	$O \prec D \mid P \mid M$	$O \prec D \mid P \mid M$	$O \prec D \mid P \mid M$	$O \prec D \mid P \mid M$
$macro - F_1^{d}$	$O \prec D \mid P \mid M$	$O \equiv D \mid P \mid M$	$O \prec D \mid P \mid M$	$O \prec D \mid P \mid M$
$micro - F_1^{d}$	$O \equiv D \mid P \mid M$	$O \equiv D \mid P \mid M$	$O \equiv D \mid P \mid M$	$O \equiv D \mid P \mid M$

Novamente, como a Tabela 5.2 mostra, o tipo de *ranking* impacta na grande maioria das métricas. Para a base de dados AT100, o desempenho do categorizador *k NN* foi impactado em 14 métricas das 19 métricas de avaliação analisadas. A análise do desempenho do categorizador *ML-k NN* mostra que este categorizador foi impactado em 6 métricas das 19 métricas. O desempenho do categorizador *VG-RAM WNN* foi impactado em 13 métricas. E, finalmente, a análise de desempenho do categorizador *VG-RAM WNN-COR* mostra que este categorizador foi também impactado em 12 métricas das 19 métricas.

A Tabela 5.3 apresenta os *rankings* mais apropriados para o tratamento de empates para cada métrica de avaliação de acordo com os experimentos (coluna *Ranking* apropriado – experimentos) e o *ranking* apropriado de acordo com a escolha deste trabalho (coluna *Ranking* apropriado – escolha deste trabalho).

Tabela 5.3 – Os *rankings* apropriados para cada métrica de avaliação.

Avaliação de conjunto	Métrica	*Ranking* apropriado – experimentos	*Ranking* apropriado – escolha deste trabalho
Ordenado	*one-error**	*D|P|M*	*M*
	coverage	*M*	*M*
	ranking loss	*O|D|P|M*	*M*
	*average precision**	*D|P|M*	*M*
	R-precision	*O|D|P|M*	*M*
Não ordenado	*Hamming loss*	*D|P|M*	*M*
	exact match	*D|P|M*	*M*
	$macro - precision^{c}$	*D|P|M*	*M*
	$micro - precision^{c}$	*D|P|M*	*M*
	$macro - recall^{c}$	*D|P|M*	*M*
	$micro - recall^{c}$	*D|P|M*	*M*
	$macro - F_1^{c}$	*D|P|M*	*M*
	$micro - F_1^{c}$	*D|P|M*	*M*
	$macro - precision^{d}$	*O|D|P|M*	*M*
	$micro - precision^{d}$	*D|P|M*	*M*
	$macro - recall^{d}$	*D|P|M*	*M*
	$micro - recall^{d}$	*D|P|M*	*M*
	$macro - F_1^{d}$	*D|P|M*	*M*
	$micro - F_1^{d}$	*D|P|M*	*M*

De acordo com a Tabela 5.3, para as métricas de avaliação do conjunto ordenado, os resultados obtidos mostram que, na maioria dessas métricas, os *rankings* Denso, Padrão e Modificados são mais apropriados do que o *ranking* Ordinal Aleatório, pois o ranking Ordinal Aleatório favoreceu os categorizadores que geraram *rankings* com empates entre as categorias de interesse. Este trabalho escolhe o *ranking* Modificado como o mais apropriado para o tratamento de empates segundo as métricas de avaliação do conjunto ordenado, pois os experimentos realizados com os categorizadores segundo a métrica *coverage* mostraram que este *ranking* foi o único que penalizou os categorizadores que produziram empates entre as categorias de interesse. Para as métricas de avaliação de conjunto não ordenado também os *rankings* Denso, Padrão e Modificados são mais apropriados do que o *ranking* Ordinal Aleatório, pois, novamente, o ranking Ordinal Aleatório favoreceu os categorizadores que geraram *rankings* com empates entre as categorias de interesse. E, dentre esses *rankings*, este trabalho escolhe o *ranking* Modificado em função da escolha para o conjunto ordenado.

Os resultados obtidos com as métricas de *precision* e *recall* mostram que, as categorias inseridas a mais no conjunto $\hat{C}_j^{|C_j|}$ penalizaram o desempenho dos categorizadores segundo a

métrica *precision* com tipos de *rankings* Denso, Padrão e Modificado, e melhoram o desempenho dos categorizadores segundo a métrica *recall* com esses tipos de *rankings*. Para a métrica *precision*, isso ocorreu porque em função das categorias inseridas a mais no conjunto $\hat{C}_j^{|C_j|}$, na média, o denominador da métrica *precision* (Equação (4.17)) foi maior do que o numerador. Para a métrica *recall* (Equação (4.21)), o denominador é o mesmo independente de empates, e o numerador, na média, foi maior do que o denominador em função das categorias inseridas a mais no conjunto $\hat{C}_j^{|C_j|}$. Note que o comportamento de diminuir *precision* e aumentar *recall*, ou aumentar *precision* e diminuir *recall* é baseado na curva de *recall x precision* [Manning08].

5.1 Trabalhos correlatos

O estudo de métricas sobre *rankings* sem empates é clássico. Entretanto, os *rankings* encontrados na prática freqüentemente possuem empates, e métricas sobre tais *rankings* têm sido pouco estudadas.

Em 1968, Cooper [Cooper68] propôs uma métrica alternativa à *precision* e *recall*, chamada Tamanho da Procura Esperada (*Expected Search Length – ESL*), para avaliar os *rankings* produzidos pelas ferramentas de buscas de documentos. Tal métrica avalia o número médio de documentos (relevantes e não-relevantes) que precisam ser examinados a partir do topo do *ranking*, gerado pela ferramenta de busca, para recuperar uma quantidade de documentos relevantes. A definição de *ESL* considera que empates no *ranking* podem existir, e os documentos empatados (mesmo grau de crença) ocupam a mesma posição do *ranking* (*ranking* Denso, ver Seção 2.2, pág. 24). Contudo, até onde pudemos examinar, a métrica *ESL* é equivalente à métrica *coverage* (ver Seção 4.1.2, pág. 69).

Fagin, em 2004 [Fagin04], propôs a técnica de agregação de *rankings* (*ranking aggregation*) para tratamento de empates no contexto de ferramentas de busca. A agregação de *rankings* consiste em, dados múltiplos *rankings*, gerados a partir da permutação dos empates existentes em um *ranking*, criar um único *ranking* que minimiza a distância Kendall-tau [Fagin03]. A distância Kendall-tau entre dois *rankings* é definida como o número de pares diferentes entre dois *rankings* [Fagin04]. Entretanto, esta técnica não pode ser empregada para avaliar o desempenho de categorizadores no domínio do nosso problema, uma vez que o

domínio do nosso problema contempla um grande número de categorias, o que possibilita inúmeros empates no *ranking* e, consequentemente, enorme número de permutações.

Em 2006 [Fagin06], Fagin definiu quatro tipos de métricas para comparar *rankings* com empates baseadas na generalização da distância Kendall-tau e na distância Spearman *footrule*, que é a soma dos valores absolutos de pares diferentes entre *rankings*, sobre duas formas de permutação. Na primeira, cada *ranking* com empates é associado a um "vetor de perfil" ("*profile vector*") e uma distância L_1 entre o *ranking* com empates e o "vetor de perfil" correspondente é definida. Na segunda, os empates existentes no *ranking* são permutados gerando múltiplos *rankings* sem empates (agregação de *rankings*). Novamente, essas métricas não podem ser empregadas para avaliar o desempenho de categorizadores no domínio do nosso problema, uma vez que o domínio do nosso problema contempla um grande número de categorias, o que possibilita inúmeros empates no *ranking* e, consequentemente, enorme número de permutações.

Este trabalho propõe uma metodologia de tratamento de empates utilizando várias formas de ordenação entre as categorias empatadas por meio dos *ranking* Denso, Padrão e Modificado, empregando as métricas de avaliação mais frequentes na literatura na avaliação de desempenho dos categorizadores multi-rótulo. Para utilizar esses *rankings*, as definições de algumas métricas foram reformuladas, tornando a definição original das mesmas mais genéricas.

5.2 Análise crítica deste trabalho

Uma das limitações deste trabalho é não correlacionar o impacto do tratamento de empates com as características das bases de dados e dos categorizadores empregados, o que poderia levar a uma abordagem mais genérica para o tratamento de empates, mais independente da base de dados e dos categorizadores. Outra possível limitação é o emprego de base de dados dentro de um único domínio de problemas de categorização. Por fim, teria sido relevante examinar outras formas de corte do *ranking* para as métricas que requerem corte. Estas limitações não foram endereçadas devido à limitação de tempo inerente ao escopo de um trabalho de pesquisa no nível de mestrado.

6 CONCLUSÃO

Neste capítulo apresentamos um sumário do trabalho, nossas conclusões e propostas de trabalhos futuros.

6.1 Sumário

Este trabalho apresenta quatro formas de *ranking* – Ordinal Aleatório, Denso, Padrão e Modificado – que oferecem tratamentos diferenciados para eventuais empates existentes entre as categorias de interesse ranqueadas por categorizadores multi-rótulo de texto automáticos. Examinamos de forma experimental o impacto de cada tipo de *ranking* no desempenho dos categorizadores *k NN*, *ML-k NN*, *VG-RAM WNN* e *VG-RAM WNN-COR* segundo as métricas de avaliação: *one-error*, *coverage*, *ranking loss*, *average precision*, *R-precision*, *Hamming loss*, *exact match*, *precision*, *recall* e F_1. Os experimentos foram realizados com duas bases de dados, contendo documentos textuais descrevendo atividades econômicas de empresas brasileiras, com características diferenciadas em termos da freqüência de ocorrência das categorias: EX100 e AT100. A base de dados EX100 contém documentos categorizados dentro de 105 categorias, onde cada categoria ocorre exatamente em 100 diferentes documentos; e a base de dados AT100 contém documentos categorizados dentro de 692 categorias, onde cada categoria ocorre em até 100 diferentes documentos.

As definições originais das métricas *one-error* e *average precision* foram reformuladas para comportar o tratamento de empates pelos *rankings* Denso, Padrão e Modificado. A reformulação dessas métricas, que chamamos de *one-error** e *average precision**, *respectivamente*, generaliza a versão original, que trata apenas os casos em que não há empates, para uma versão que trata empates. As definições originais das demais métricas não foram alteradas, pois permitem trabalhar com *rankings* utilizados neste trabalho.

Nossos resultados experimentais mostraram que, por causa de empates, o tipo de *ranking* empregado afeta significativamente o desempenho dos categorizadores utilizados segundo as métricas examinadas.

6.2 Conclusões

Os resultados experimentais apresentados no Capítulo 4 mostram que o desempenho de um categorizador segundo uma determinada métrica é significativamente diferente (teste t pareado bicaudal com nível de significância 5%) dependendo do tipo de *ranking* empregado. Os experimentos realizados com a base de dados EX100 mostram que, o desempenho dos categorizadores $k\,NN$, $ML\text{-}k\,NN$, *VG-RAM WNN* e *VG-RAM WNN-COR* é impactado pelo tipo de *ranking* empregado em 16, 12, 12 e 10 métricas, respectivamente, das 19 métricas de avaliação analisadas neste trabalho. E os experimentos realizados com a base de dados AT100 mostram que, o desempenho dos categorizadores $k\,NN$, $ML\text{-}k\,NN$, *VG-RAM WNN* e *VG-RAM WNN-COR* é impactado pelo tipo de *ranking* empregado em 14, 6, 13 e 12 métricas, respectivamente, das 19 métricas de avaliação analisadas neste trabalho.

Na maioria das métricas analisadas, os *rankings* Denso, Padrão e Modificado são mais apropriados para tratamento de empate do que o *ranking* Ordinal Aleatório, pois o ranking Ordinal Aleatório favoreceu os categorizadores que geraram *rankings* com empates entre as categorias de interesse. Os experimentos realizados com os categorizadores segundo a métrica *coverage* mostraram que o *ranking* Modificado foi o único que penalizou os categorizadores que produziram empates entre as categorias de interesse. Então, este trabalho demonstra que o *ranking* Modificado é o mais apropriado para tratamento de empates entre categorias de interesse nos *rankings* para as métricas: *one-error**, *coverage*, *ranking loss*, *average precision**, *R-precision*, *Hamming loss*, *exact match*, $macro-precision^{c}$, $micro-precision^{c}$, $macro-recall^{c}$, $micro-recall^{c}$, $macro-F_1^{c}$, $micro-F_1^{c}$, $macro-precision^{d}$, $micro-precision^{d}$, $macro-recall^{d}$, $micro-recall^{d}$, $macro-F_1^{d}$ e $micro-F_1^{d}$.

6.3 Trabalhos futuros

Os resultados satisfatórios obtidos neste trabalho motivam continuar as pesquisas de definição de métricas e formas de ranqueamento para tratamento de empates existentes nos *rankings* produzidos pelos categorizadores multi-rótulo de texto.

Um estudo importante é correlacionar o impacto do tratamento de empates com as características das bases de dados empregadas, o que poderia levar a uma abordagem mais genérica para o tratamento de empates, mais independente da base de dados.

Outro estudo a ser realizado é utilizar outras bases de dados de problemas de categorização de domínio diferente do utilizado neste trabalho. Este estudo é importante, pois permite verificar o impacto do tratamento de empates no desempenho dos categorizadores segundo as métricas de avaliação em bases de dados com características diferentes das empregadas neste trabalho.

Outro estudo relevante que dever ser realizado é utilizar outras formas de corte no *ranking* para as métricas que requerem corte. Este estudo permite verificar o impacto do tratamento de empates no desempenho dos categorizadores com políticas de cortes que diferem da política de corte ideal adotada neste trabalho.

7 REFERÊNCIAS BIBLIOGRÁFICAS

[Aiolli08] F. Aiolli, R. Cardin, F. Sebastiani, and A. Sperduti. *Preferential text classification: learning algorithms and evaluation measures*. Information Retrieval Journal, pages 1386-4564, 2008.

[Aleksander98] I. Aleksander. *RAM-Based Neural Networks*, chapter From WISARD to MAGNUS: a Family of Weightless Virtual Neural Machines, pages 18–30. World Scientific, 1998.

[Antiqueira05] L. Antiqueira. Obtenção e Associação de Termos na Construção de uma Ontologia para a Área de Nanotecnologia. São Carlos: USP, 2005. 40 p. Monografia de Graduação – Instituto de Ciências Matemáticas e de Computação, USP, São Carlos, 2005.

[Aspell08] ASPELL. *GNU Aspell*. Disponível: http://aspell.net/ . Último acesso em: 20 de Agosto de 2008.

[Badue08] C. Badue, F. Pedroni, and A. F. De Souza. *Multi-Label Text Categorization using VG-RAM Weightless Neural Networks*. Proceedings of the 10th Brazilian Symposium on Neural Networks (SBRN'08), pp. 105-110, Salvador, Bahia, Brazil, October 2008.

[Baeza99] R. Baeza-Yates, and B. Ribeiro-Neto. *Modern Information Retrieval*. 1. ed. New York: Addison-Wesley, 1999.

[Baoli03] L. Baoli, Y. Shiwen, and L. Qin. *An Improved k-Nearest Neighbor Algorithm for Text Categorization*. In Proceedings of the 20th International Conference on Computer Processing of Oriental Languages, Shen Yang, China, pages 469-475, 2003.

[Boutell04] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown. *Learning multi-label scene classification*. Pattern Recognition, 37(9): pages 1757–1771, 2004.

[Cherman07] E. Cherman, H. de Lee, D. Honorato, C. Coy, J. Fagundes, J. Góes, F. Wu. Metodologia de mapeamento automático de laudos colonoscópicos. XVI EAIC, 2007.

[Ciarelli08] P. M. Ciarelli. Rede Neural Probabilístca para a Classicação de Atividades Econômicas. Vitória: UFES, 2008. 82 p. Dissertação – Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal do Espírito Santo, Vitória, 2008.

[Ciarelli09] P. M. Ciarelli, E. Oliveira, and C. Badue. *Multi-Label Text Categorization Using a Probabilistic Neural Network*. International Journal of Computer Information Systems and Industrial Management Applications (IJCISIM), July 2009 (accepted for publication).

[Clare01a] A. Clare and R. D. King. *Knowledge discovery in multi-label phenotype data*. In Lecture Notes in Computer Science 2168, L. D. Raedt and A. Siebes, Eds. Berlin: Springer, pages 42–53, 2001.

[Clare01b] A. Clare and R. D. King. *Knowledge discovery in multi-label phenotype data*. In Lecture Notes in Computer Science, volume 2168, pages 42–53, 2001.

[CNAE03] CNAE. Classificação Nacional de Atividades Econômicas – Fiscal (CNAE-Fiscal) 1.1. Instituto Brasileiro de Geografia e Estatística (IBGE), Rio de Janeiro, RJ, 2003.

[Comité03] F. D. Comité, R. Gilleron, and M. Tommasi. *Learning multi-label alternating decision tree from texts and data*. In Lecture Notes in Computer Science, volume 2734, pages 35–49. Springer, 2003.

[Cooper68] W.S Cooper. *Expected search length: A single measure of retrieval effectiveness based on weak ordering action of retrieval systems*. Journal of the American Society for Information Science, 19(1), pages 30 – 41, 1968.

[Crowell03] J. Crowell, Q.T. Zeng, S.Kogan. *A Technique to Improve the Spelling Suggestion Rank in Medical Queries*. AMIA 2003 Symposium Proceedings, page 823, 2003.

[DeSouza07] A. F. De Souza, F. Pedroni, E. Oliveira, P. M. Ciarelli, W. F. Henrique, and L. Veronese. *Automated free text classification of economic activities using vg-ram weightless neural networks*. In 7th IEEE International Conference on Intelligent Systems Design and Applications, pages 782–787. IEEE Computer Society, 2007.

[DeSouza08] A. F. De Souza, C. Badue, B. Z. Melotti, F. T. Pedroni, and F. L. L.

Almeida. *Improving vg-ram wnn multi-label text categorization via label correlation*. In 2nd Workshop on Intelligent Text Categorization and Clustering (WITCC'08), 8th IEEE International Conference on Intelligent Systems Design and Applications (ISDA'08), volume 01, pages 437–442. IEEE Computer Society, 2008.

[DeSouza09a] A. F. De Souza, F. Pedroni, E. Oliveira, P. M. Ciarelli, W. F. Henrique, L. Veronese, and C. Badue. *Automated Multi-label Text Categorization with VG-RAM Weightless Neural Networks*. Neurocomputing, vol. 72, no. 10-12, pp. 2209-2217, June 2009.

[DeSouza09b] A. F. De Souza, B. Z. Melotti, and C. Badue. *Multi-Label Text Categorization with a Data Correlated VG-RAM Weightless Neural Network*. International Journal of Computer Information Systems and Industrial Management Applications (IJCISIM), July 2009 (accepted for publication).

[Dunlop97] M. D Dunlop. *Time Relevance and Interaction Modeling for Information Retrieval*, in Proc. ACM SIGIR, pages 206-213, 1997.

[Elisseeff02] A. Elisseeff and J. Weston. *A kernel method for multi-labelled classification*. In Advances in Neural Information Processing Systems, volume 14, pages 681–687. MIT Press, 2002.

[Fagin03] R. Fagin, R. Kumar, and D. Sivakumar. *Comparing top k lists*. Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms, pages 28–36, Philadelphia, USA, 2003.

[Fagin04] R. Fagin, R. Kumar, M. Mahdian, D. Sivakumar, and E. Vee. *Comparing and aggregating rankings with ties*. Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, pages 47-58, France, 2004.

[Fagin06] R. Fagin, R. Kumar, M. Mahdian, D. Sivakumar, & E. Vee. *Comparing partial rankings*. SIAM Journal on Discrete Mathematics, 20(3), pages 628–648, 2006.

[Gao04] S. Gao, W. Wu, C.-H. Lee, and T.-S. Chua. *A MfoM learning approach to robust multiclass multi-label text categorization*. In Proceedings of the 21st International Conference on Machine Learning, pages 329–336, 2004.

[Hair05] J. F. Hair, R. E. Anderson, R. L. Tatham e W. C. Black. Análise Multivariada de Dados. Tradução por Adonai Schlup Sant'Ana e Anselmo Chavese Neto. Quinta Edição. US, 2005.

[Hao07] X. Hao, X. Tao, C. Zhang. *Yunfa Hu, An Effective Method To Improve kNN Text Classifier*. Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2007. SNPD 2007. Eighth ACIS International Conference, vol.1, no., pages 379-384, July 30 2007-Aug. 1 2007.

[Haykin99] S. Haykin. Redes Neurais Princípios e práticas. 2ª Edição. São Paulo, 1999.

[Hull93] D. Hull. *Using statistical testing in the evaluation of retrieval experiments*. Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval, pages 329-338, USA, 1993.

[Kazawa05] H. Kazawa, T. Izumitani, H. Taira, and E. Maeda. *Maximal margin labeling for multi-topic text categorization*. In Advances in Neural Information Processing Systems 17, pages 649–656. MIT Press, 2005.

[Ludermir99] T. B. Ludermir, A. C. P. L. F. Carvalho, A. P. Braga, and M. D. Souto. *Weightless neural models: a review of current and past works*. Neural Computing Surveys, 2: pages 41–61, 1999.

[Manning08] C. D. Manning, P. Raghavan, and H. Schútze. *An Introduction to Information Retrieval*. Cambridge University Press, Cambridge, England, 2008.

[Martins04] D. Martins, and M. J Silva. *Spelling Correction for Search Engine Queries*. In Book Series of Lecture Notes in Computer Science, Vol. 3230, pages 372-383, 2004.

[McCallum99] A. McCallum. *Multi-label text classification with a mixture model trained by EM*. In Working Notes of the AAAI'99 Workshop on Text Learning, pages 1–7, 1999.

[Mitchell97] T. M. Mitchell. *Machine learning*. McGraw Hill, New York, US, 1997.

[Mitchell98] R. J. Mitchell, J. M. Bishop, S. K. Box, and J. F. Hawker. *RAM-Based Neural Networks, chapter Comparison of Some Methods for Processing Grey Level Data in Weightless Networks*, pages 61–70. World

Scientific, 1998.

[Monard03] M. C. Monard & J. A. Baranauskas. Conceitos sobre Aprendizado de Máquina. In Sistemas Inteligentes – Fundamentos e Aplicações, , S.O Rezende, Editora Manole, pages 89-114, 2003.

[Oliveira08a] E. Oliveira, P. M. Ciarelli, A. F. De Souza, and C. Badue. Using a Probabilistic Neural Network for a Large Multi-Label Problem. Proceedings of the 10th Brazilian Symposium on Neural Networks (SBRN'08), pp. 195-200, Salvador, Bahia, Brazil, October 2008.

[Oliveira08b] E. Oliveira, P. M. Ciarelli, and C. Badue. *A Comparison Between a kNN based Aproach and a PNN Algorithm for a Multi-Label Classification Problem*. Proceedings of the 2nd Workshop on Intelligent Text Categorization and Clustering of the 8th International Conference on Inteligent System Design and Applications (ISDA'08), pp. 628-633, Kaohsiung City, Taiwan, November 2008.

[Picard84] R. R. Picard, and R. D. Cook. *Cross-Validation of Regression Models*. Journal of the American Statistical Association, 79(387), pages 575–583, 1984.

[Rijsbergen79] V. Rijsbergen, C. J. *Information Retrieval* (Second ed.). Butterworths, London, UK, 1979. Available at http://www.dcs.gla.ac.uk/Keith.

[Romero04] E. Romero, L. Màrquez, and X. Carreras. *Margin maximization with feed-forward neural networks: a comparative study with svm and adaboost*. Neurocomputing, 57: pages 313–344, 2004.

[Salton75] G. Salton, A. Wong, and C. Yang. *A vector space model for automatic indexing*. Communications of the ACM 18, 11, 613–620. Also reprinted in [Sparck Jones and Willett 1997], pages 273–280, 1975.

[Sbc09] Sociedade Brasileira de Computação, Grandes desafios da pesquisa em computação no Brasil 2006-2016. Último acesso em 12/08/2009.

[SCAE08] Sistema Computacional de Codificação Automática de Atividades Econômicas (SCAE), Projeto de Classificação Automática em CNAE-Subclasses – Relato de Cumprimento de Metas No. 4. Universidade Federal do Espírito Santo, Vitória, 2008.

[Schapire99] R. E. Schapire and Y. Singer. *Improved boosting algorithms using*

confidence-rated predictions. Machine Learning, 27(3): pages 297–336, 1999.

[Schapire00] R. E. Schapire and Y. Singer. *BoosTexter: a boosting-based system for text categorization*. Machine Learning, 39(2/3): pages 135–168, 2000.

[Sebastiani02] F. Sebastiani. *Machine learning in automated text categorization*. ACM Computing Surveys, 34(1): pages 1–47, 2002.

[Sparacino00] G. Sparacino, C. Tombolato, C. Cobelli. *Maximum-likelihood versus maximum a posteriori parameter estimation of physiological system models: the c-peptide impulse response case study*. Biomedical Engineering, IEEE Transactions on Volume 47, Issue 6, pages 801 – 811, June 2000.

[Student08] Student. *The Probable Error of a Mean*. Biometrika on Volume 6, pages 1 – 25, 1908.

[Ueda03] N. Ueda and K. Saito. *Parametric mixture models for multi-label text*. In Advances in Neural Information Processing Systems, volume 15, pages 721–728. MIT Press, 2003.

[Wikipedia09] Wikipedia. http://en.wikipedia.org/wiki/*Ranking*. Último acesso em 12/08/2009.

[Witten05] Ian H. Witten & E. Frank. *Data Mining Practical Machine Learning Tools and Techniques*. Second Edition. US, 2005.

[Yang99] Y. Yang. *An Evaluation of Statistical Approaches to Text Categorization*. In Information Retrieval, Volume 1, pages 69-90, Hingham, US, 1999.

[Yang01] Y. Yang. *A study of thresholding strategies for text categorization*. In Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'01), pages 137–145, New Orleans, Louisiana, United States, 2001.

[Yavuz98] T. Yavuz and H. Altay Guvenir. *Application of k-nearest neighbor on feature projections classifier to text categorization*. Proceedings of ISCIS, 13th International Symposium on Computer and Information Sciences, pages 135-142, 1998.

[Zhang06] M.-L. Zhang, Z.-H. Zhou,. *Multi-label neural networks with applications to functional genomics and text categorization*. IEEE

Transactions on Knowledge and Data Engineering 18(10) , pages 1338–1351, 2006.

[Zhang07] M.-L. Zhang and Z.-H. Zhou. *ML-KNN: A lazy learning approach to multi-label learning*. Pattern Recognition, 40(7): pages 2038–2048, 2007.